Knowledge-Based System Approach to Integrated Design of Multistorey Office Buildings at the Preliminary Stage

Mathi Ravi

A Thesis

in

the School for Building

Presented in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy at

Concordia University
Montréal, Québec, Canada

February 1998

0-612-40307-6

ABSTRACT

Knowledge-Based System Approach to Integrated Design of Multistorey Office Buildings at the Preliminary Stage

Mathi Ravi, Ph. D.
Concordia University, 1998

The work described in this thesis represents a significant original contribution in identifying, acquiring, synthesising and organising design knowledge related to tall buildings. Due to the non-formalised nature of preliminary design, no computer-based tools could be developed using common procedural programming methods. Preliminary design stage also has the maximum ability to positively impact the final cost of the building project. In the absence of computer-based tools and also due to resource and time constraints, the current preliminary design tasks are mostly restricted to exploring one or two design alternatives often proceeding with a less than efficient solution. Therefore a computer-based design tool that could be used by designers at the initial stages of a project to explore many design alternatives on a comparative basis could potentially lead to better and more economical designs than those produced without such a design tool. Preliminary design needs the consideration of all relevant parameters affecting the product designed. For multistorey buildings the architectural and structural design considerations are closely tied together and in this work an integration of the two is demonstrated. Architectural planning considerations for office buildings and feasible structural solutions are incorporated in a knowledge-base. A model for the design system is developed that incorporates the gathered design knowledge in declarative form which is then used to selectively generate and evaluate integrated building design solutions. Thus part of this work is the implementation of a knowledge-based system, Tall-D, using a development tool. Then, a methodology is devised for the testing of Tall-D under practical conditions. External industry experts are invited to furnish completed project designs and evaluate the corresponding designs performed by Tall-D by answering a comprehensive questionnaire. The result is a very favourable endorsement of the many facets of the current work as well as that of Tall-D system performance and features. The overall thrust of this work has been to research the area of preliminary design with a view to originating computer-based tools for this phase in engineering.

Acknowledgements

I sincerely thank Dr. Claude Bédard, Professor, School for Building and Dean, School of Graduate Studies, for supervising this work. He was thorough in his review of the thesis and provided valuable input all along the duration of this work. Though we did not always agree on issues, he always acted in the best interests of the research project. He was kind and understanding in many difficult situations.

Funding for of my doctoral research project was provided by Natural Sciences and Engineering Research Council of Canada as well as FCAR, Quebéc. I thank Dr. Bédard and Dr. Paul Fazio, Director, School for Building, for making available these funds for this work. I was also fortunate to tutor many courses in School for Building (previously known as Centre for Building Studies) as well as work as part-time teacher in the Department of Computer Science. Both provided additional funds and valuable experience. I thank Dr. T. Stathopoulos and Dr. V. S. Alagar of the respective departments for extending me such assistance. I also thank School for Building and the Faculty of Engineering, Concordia University for granting me the fee waiver award and graduate studies bursary.

I would like to thank the thesis committee members Dr. T. Radhakrishnan, Prof. C. Marsh, Dr. Kinh Huy Ha, Dr. O. Moselhi and Dr. S. Alkass for their suggestions and approval of the thesis proposal. I would like to thank Mr. Serge Vézina, the structural engineer and Mr. Antonio Niro, the architect, both of Montreal, for their time and efforts in the validation of Tall-D, the knowledge-based system developed as part of this work.

I like to take this opportunity to thank my parents Mr. Mathi Subba Rao and Mrs. Mathi Vasundara Devi for taking great interest in my welfare and progress. I will always be indebted to them for their sacrifices and cannot return their affection in full measure. I dedicate this work to them. Similar sentiments are due in describing rest of the family - Vani, Kumar, Chandrasekar and Anupama. My little nieces Haripriya and Himapriya keep me entertained from around the globe and are always in my thoughts.

I thank the many faculty, graduate students and staff of the School for Building and others among the larger Concordia community as well as the East Indian community of Montreal for their empathy, interaction and friendship. Together they made Montreal a home away from home. Of course one would feel at home in Montreal anyway, the exceptional cosmopolitan city it is. Of particular mention either of families or individuals from Montreal and Ottawa Valley areas whose company I enjoyed are Atul and Archana, Baskaran and Shanthi, Chander and Vidya, Chandra and Monica, Frank and Mary-jane, Gerard and Elizabeth, Gowri and Vasanthi, Jayakumar and Banu, Kannan (aka Subbu), Kannan and Lakshmi, Mohan and Sreemathi, Prasad, Pritham and Mythili, Ragu and Sandhya, Ramakrishna and Sandhya, Ramalingom and Priya, Ramesh and Jaya, Ramesh and Jayshree, Reddy and Anuradha, Srinivas, Subbu and Laavanya, Sohan and Pushpa, Sunder and Rama, Surej and Usha, Suresh and Sindhu, Venkatesh, Vijay and Anjani, and Zaheeruddin.

I thank my Manager at AECL, Mr. Rick Didsbury, Engineering Tools Branch, for his co-operation during the final stages of thesis writing. I would also like to thank Kannan, Ramnik, Ramakrishna, Sharma, Jeff, Ted and Bruce among others at AECL for their moral support and in a few cases, for accommodating my work schedule where needed. In my current home - Ottawa Valley - many community members have been supportive and I thank them all.

Deep River, Ontario.
Feb. 1998

TABLE OF CONTENTS

Chapter | Page

TABLE OF FIGURES

Figure | Page

LIST OF TABLES

Table Page

CHAPTER 1

Introduction

1. 1 General

The designers of buildings have profitably used the computer in routine design office chores such as structural analysis and drafting over the last three decades. This was the result of increasing familiarity with computer applications, decreasing hardware costs (since the 80s) as well as the availability of a wide range of application software. The driving force was the need to increase both productivity and accuracy in designs. A large percentage of architectural/engineering firms now have some data-processing equipment. Even the smallest firms are acquiring computers and those with modest systems are upgrading theirs. Yet the use of the computer aids so far is restricted to applications which can be characterised as detailed design, whereas the potential to improve both the design process and the resulting buildings is greatest at the preliminary design stage. The reason is the major influence that preliminary design decisions have on the overall performance as well as on the life-cycle costs of the building project. There are only few computer-based systems available to assist in these early design tasks as opposed to the extensive availability of tools for structural analysis, member design, detailing etc. The reason is that the preliminary design process is typically a not so well-defined procedure, making consequently conventional programming techniques not suitable to automate such a process.

The knowledge-based system (KBS) approach, a derivative of research in the area of Artificial Intelligence, provides a practical technique for computer-based decision-making in less formalised areas, such as preliminary design. In general, applications of the KBS approach to building engineering problems have focused on narrow domains. The reason for this was to ensure robust system performance with relative ease given the scarcity of affordable software environments capable of developing large knowledge-based systems. However, preliminary building design by nature encompasses multi-disciplinary design decisions. At an early design stage, the designer has considerable influence on the building subsystems and each design decision affects various subsystems. In computer-aided building design, *integration* therefore holds the key for improving the design process. There are few computer-based design systems however that attempt to address this need for integration in automated building design.

The KBS developed and described in this report is an experiment in simulating integrated building design and consists of modules capable, on the one hand, of assembling overall building configurations, and on the other, of developing compatible lateral and gravity load-resisting structural sub-systems. These two areas of building design are representative of the architectural and structural domains, and are considered in an integrated way in the proposed computer-based design system.

1. 2 Motivation and Relevance

The motivation for the current work comes from the fact that there is great potential for taking advantage of computers in the work place including the design office. Computers are basically used for design at the detailed stage or for drafting. However there is a lack of computer-based applications at the preliminary stages as already mentioned, where there is potential for the greatest gains and impact on in the final design and throughout the life-cycle. A research project that incorporates the characteristics of preliminary design in order to develop a computer-based system that encompasses the different disciplines in a practical situation, will contribute to realising some of the unfulfilled gains in an enhanced preliminary design process.

There are numerous examples of inadequate performance in buildings that can be attributed to deficiencies in design and construction. There is major scope for improving the design process at the preliminary stages of design. Moreover designers can no longer consider the different components and systems in buildings in isolation. An integrated approach that spans horizontally among the different building engineering disciplines and vertically in increasing detail down to the component level would be highly desirable. Due to the complexity of such an approach and the practical limitations of manual design, automation using computers is an obvious choice. As mentioned above, at the preliminary design stage, the use of KBS techniques is promising to realise such design automation. Hence in this work, an attempt is made to contribute towards improvement of the design process by developing a computerised system that follows the KBS approach. The reasons for selecting the preliminary phase of integrated building design as the domain and to implement it as a KBS are many. Primary among them are:

a) The *preliminary* phase has a greater potential for achieving significant design improvements than the later design phase;
b) Computer-based tools are not currently available to perform integrated design tasks; an attempt is made in this work to account for *integration* issues as these relate to architectural and structural domains;
c) The choice of the KBS approach as the means of exploring computer-based integrated design follows from the nature of preliminary design, which is a not well formalised process, i.e. not suitable for a procedural programming language.

1. 3 Objectives and Limitations

1.3.1 Objectives of the Work

The main objective of the current work, which follows from the motivation above, is to improve design practice by developing a computer-based system capable of accounting for preliminary design as well as integration between disciplines. The objectives of the work can be itemised as follows:

a) To contribute in the formalisation of a significant portion of knowledge in the area of structural layouts and structural elements design of tall buildings.

b) To establish a design process model for the integrated design of buildings at the preliminary stage by investigating the issues related to the two areas of structural and architectural design;

c) To develop an integrated design tool, the Tall-D system, that can be used to explore different alternatives for multistorey buildings; and

d) To demonstrate the applicability of the knowledge-based systems approach to implement the above integrated design model by means of an object-based model of building and components, and with a rule-based declaration of the design knowledge.

1.3.2 Limitations of the System

The system considers for integrated design purposes the two domains of architectural and structural design, and treats these as representative of two interacting fields of expertise in design decisions.

The limitations of the proposed preliminary design system are listed below:

a) The computer-based system considers only rectangular floor plans with office occupancy, and only one typical storey layout being used over the height as the basis of structural design.

b) Building height is restricted to the equivalent of fifty storeys and no consideration is given to the substructure and to underground levels, nor to the presence of a plaza.

c) Concrete or steel construction is considered. Composite construction which is also employed in high-rise construction is considered only where a concrete core and steel perimeter columns occur. Composite columns and composite shearwall construction are not covered in Tall-D.

d) Energy consumption and thermal comfort considerations which are dependent on the HVAC system, the envelope, the building thermal mass, building shape and orientation also influence decision making at the preliminary design stage. Only the building shape factor is included, however, in the current work.

1.4 Layout of Thesis

In Chapter 2, the building design process and knowledge-based design systems are presented. A discussion about the nature of the design process as well as the role of architects and engineers as design partners is included. The different components in typical KBS, the solution of problems by knowledge-base systems, a review of related work in KBS for building design applications as well as integrated design are presented.

In Chapter 3, the overall approach used in the current system in terms of tools and the sources of information is described. A model of the design process implemented in the system is also presented.

In Chapter 4, the development and evaluation of overall building configurations is presented. Approximate cost estimation of the design alternatives is also discussed in this chapter.

In Chapter 5, the spatial configuration of structural subsystems and the approximate sizing methods are described. Spatial configuration follows the type of structure selected. Approximation of loads and in some cases the resulting forces and corresponding approximate scaling of structural members are also presented.

In Chapter 6, the application and validation of system Tall-D is presented using three case studies, in collaboration with two expert designers. The study highlights similarities and differences between the system-generated solutions and those produced in practice. This study also led to a qualitative evaluation by the experts of the performance of the system by means of a questionnaire.

In the final chapter, concluding remarks along with contributions to the state-of-the-art in computer-aided building design are discussed as well as scope for further research.

CHAPTER 2

Review of Previous Work on Building Design and Knowledge-based Design Systems

A review of the design process is presented in the first half of this chapter. Issues such as the different factors in the design of multistorey buildings, different models of the design process as well as the building as a product are presented. In the later half knowledge-based systems related to buildings are reviewed.

2. 1 Factors in the Design of Multistorey Buildings

2. 1. 1 General

Design is the development of the physical description of an artifact, subject to a given set of constraints or specifications. Design evolves and cycles over time, subject to a variety of constraints such as time and money. There are two major phases in the building design process. Preliminary design involves the development of concepts and the selection of overall best configurations and subsystems. Schematic representation of alternatives is produced. Detailed design constitutes the elaboration of the subsystems decided upon earlier. It defines a complete solution down to the final detail. Analysis and optimisation are often part of the detailed design stage (Radford and Gero 1988). It results in drawings with all information such as the structural, architectural, mechanical, material, geometric and interface descriptions of the components in the building. Generally there is no single solution with optimal performance with respect to the different requirements due to sometimes conflicting objectives. Thus the selection of a suitable solution comprised of various building components or configurations often involves evaluation on a subjective scale.

In addition to sculptural shapes at the top and elaborate detailing at the base, tall buildings generally have either a dramatically shaped mass or variations of historical styles (Taranath 1988). Factors that influence design decisions in multistorey buildings are the different loads, structural framing, mechanical services, vertical transportation and life safety systems (Guise 1990). Many designers consider design as being intuitive as well and therefore not entirely logical. On occasions additional investigators are required due to unique or creative design proposals (Holgate 1986). To account for such intuitive thinking "creative" or "innovative" design systems may be developed (Cagan and Agogino 1989). However it is an area of research in itself that still requires significant advances in order to develop realistic applications. The target area of research in studying preliminary design and developing the Tall-D system is for the "routine" design of multistorey buildings.

The primary concern of structural designers in multistorey building design is to achieve an optimal lateral load-resisting system. Factors to be considered are the building slenderness, the intensity of loading, floor to floor height, gravity load-resisting structural system, horizontal and vertical distribution of services, flexibility of the occupant space, location of service core as well as conformance to Code regulations such as structural strength and serviceability, fire safety, travel distance, zoning laws etc. Some of the above issues are not exclusive concerns for the engineer, but often dominant ones for the architect. In buildings taller than forty storeys, the choice of a lateral structural system is a critical factor in ensuring an economical and functional design (Taranath 1988). The choice in the case of buildings with less than forty storeys, though of lesser consequence, requires the services of an expert structural designer to propose and evaluate optimal solutions. The implications of lateral load resistance in the context of tall buildings is further discussed in section 5.1.

2. 1. 2 Roles of Architects and Structural Engineers

Traditionally the architect was the master builder, with control over the entire design and building processes. Being not only responsible for the aesthetics, he was also responsible for the construction. Industrialization and increasingly complex projects meant that the architect had to abandon areas of activity that were better served by expert engineers. Such areas in building design and construction include the structural, mechanical, electrical systems and construction engineering. In addition, services of experts in value engineering and finance are also required (Holgate 1986).

Thus the bigger the project, the bigger the project team. The design is therefore the result of a gradually evolving concept from an initial scheme generated by the architect and the owner of the building. The initial concept may be influenced by the requirements of a major building occupant as well as those of the owner. Further developments of the concept in terms of required floor area, functionality, site conditions, zoning laws, finances as well as the architect's desire to create a distinct impact eventually lead to schematic architectural drawings.

Upon approval from civic authorities, more detailed designs are generated. With the help of engineers from different disciplines, the major subsystems are sufficiently detailed for each participant to have information regarding the others' requirements and responsibilities, therefore enabling everyone to work on their respective subsystem to finalise details. This phase is traditionally the detailed design stage and is coordinated by periodic meetings among the different disciplines of the design team which communicate with drawings and documents.

Thus the role of the architect in the building design process spans the entire gamut of operations ranging from aesthetics to construction. However project managers are gaining more and more control over the construction operations.

To develop a model for a computer-based automated system with an emphasis on integration and preliminary stages of design, the traditional role of the architect in the building design process is used as a basis. The control mechanism of such a system simulate his role in bringing about an initial concept, and then incorporate the input from the different specialists of the design team. Such a model is presented and implemented as a knowledge-based system called hereafter the Tall-D system (for Tall building Design). The building as an artifact is modelled with a hierarchy of objects representing the overall building configuration, floor plan, service core, structural systems (gravity and lateral load resisting), structural loads etc. The process of design is laid down in declarative form in a rule-base arranged into different sets corresponding to specific design tasks. The two domains used to simulate integrated design are architectural planning and structural system configuration. The model provides for further expansion by the inclusion of other subsystems not yet included such as the envelope, HVAC and other services. The details of the Tall-D system are presented in Chapter 4. The system serves as an experimental model for the study of integrated design issues where different constraints are considered in preliminary design. The resulting tool has elicited good response from designers experienced in the design of multistorey buildings regarding its utility for preliminary design, as presented in the penultimate chapter.

2. 2 Preliminary Design

Preliminary design is that phase of the building design process which starts with an initial concept and proceeds through the investigation of major details with many different alternatives. It is also the design phase when decisions have the potential to effect maximum influence on the final design and project cost (Construction Industry Report 1987, Steyert 1972). Designers however tend to spend most of their working time on the detailed design phase, where the scope for significant improvement is much less. They are often able to generate only a single solution, or at the most a few that satisfy design criteria. In traditional design practice, constraints on time and design costs (and fees) force designers to such a path. Extensive generation and evaluation of alternatives is only possible with the help of computer-based methods.

Computer applications for preliminary design have not been commercialized yet, though recent studies of work in-progress are reviewed in two subsequent sections in this chapter. One of the reasons for this situation is that preliminary design has not yet evolved into a well-defined procedure (Rychener 1989). Consequently, the popular procedural methods of computing cannot satisfactorily automate an ill-defined design domain such as preliminary design. However research in knowledge-based systems (KBS) has yielded an approach that can be used in solving such design problems.

The KBS approach is suited to deal with the less formalised nature of preliminary design tasks that are better represented by means of the declarative programming paradigm. Available facilities in KBS development tools also reduce the development effort as compared to the use of procedural programming methods and languages to automate similar tasks. The application of the KBS approach to engineering design problems is reviewed in section 2. 5.

2. 3 Integrated Design

2. 3. 1 General

Fragmentation in the construction industry is the phenomenon whereby communication and coordination problems arise between different groups working on the same project. Fragmentation over time and between disciplines affects the final design. Duplication of design effort, errors and loss of efficiency in design follow. Taking advantage of the data processing and communication capabilities of computers can vastly improve the situation by integrating the groups and processes currently subject to fragmentation. To concentrate on preliminary design, is to recognise the importance of this design phase since preliminary design has the maximum influence on the major design parameters. When performing preliminary design, an overall view of the design process and design artifact is needed. It takes an expert with broad-based knowledge. In other words, the designer at the early stages needs to understand the many factors and actors affecting the building being designed. Such an *integrated* approach to building design must encompass the structural, envelope, mechanical and interior subsystems as well as their interactions and influence on one another (Rush 1986).

However, integration remains hard to achieve in building design particularly because of the lack of agreement or of an evolved standard across the industry concerning an integrated model for facilities such as buildings. Until recently, the scarcity of computer-based tools to model facilities beyond drafting and to support technical design applications made the development of integrated applications very difficult. Advances in different branches of computing such as new programming paradigms, hybrid knowledge-based systems etc. have however resulted in better instruments for experimenting with realistic models for integrated design of built facilities.

2. 3. 2 Computer-based Integration of Design

Currently integration in building design can only be achieved through the concerted efforts of the various design teams working on a given project. With the increasing use of computers, the architectural and engineering design communities have taken steps towards the development of integrated computer-based design approaches (BSAI 1991, Menniti 1996). These efforts have shown that architectural aesthetics, efficient use of space, functionality and life-cycle costs of the building are affected by interactions between the main subsystems composing a building.

Some broad-based standardisation initiatives that strive for integration in computer-based design tools are described below and discussion of specific research projects in integrated design is presented in section 2.5.3.

The most significant international effort that is likely to lead to software tools ultimately facilitating integration in computer-aided design, computer-aided engineering and computer-aided manufacture (CAD/CAE/CAM) areas is the ISO-STEP standardisation effort (ISO 1994), Standard for the Exchange of Product Model Data (STEP), which is officially known as the ISO Standard 10303, and the development effort is lead by the

ISO Technical Committee on Industrial Data and Processes. This standard when complete will have a comprehensive description of products in the form of *data models* with each industry or discipline having its own specific version. The reasoning behind this initiative is to provide for data that encapsulates not only the geometry but also the design intent and makes it exchangeable between discipline specific applications as well as makes it available throughout the life cycle of the product. STEP is supported by various governmental organisations and industries around the world.

An industry initiative that began in 1993, working to improve utility of CAD and other software across the entire life-cycle of AEC projects and facilities management is lead by the International Alliance for Interoperability (IAI) (Herold 1997). The motivation for this initiative was essentially the lack of continuum between the different phases of a construction project. As the different phases and disciplines of a project use different software, there is loss of critical data in transit between phases and between disciplines. The result is loss of productivity despite use of computers. IAI determined that there is a need for a set of standardised object definitions that would retain the critical information through any application that had been developed to use those standard definitions. The new standardised objects are called industry foundation classes (IFC). IFC are a library of commonly defined objects that create intelligent project data. Due to the similarity between the objectives of IAI and STEP, a formal liaison with the STEP AEC committee was established in 1996. IFC may lead to quicker release of software from CAD vendors supporting it, however is not so comprehensive nor complex as STEP, the latter effort encompassing all industry as opposed to the AEC focus of IFC library. STEP has several industry-oriented standards or Application Protocols (APs) under development. STEP promises to be a better solution in the long term as it promotes data standards that facilitate cross discipline exchange of data and is going through rigorous development with wide-ranging participation.

Electronic Data Interchange is an established industry data exchange format used in many business operations such as purchasing (EDI 1997). It is gradually being used in CAD systems to transfer parts information to suppliers. A United Nations organisation, EDIFACT has been established to develop EDI standards. This initiative is more oriented to electronic commerce than engineering and design as is the case with STEP.

A standard for exchange of information has been developed for the US army to support parts and systems acquisition, configuration management, maintenance and life-cycle support. Known as the Continuous Acquisition and Life Cycle Support (CALS) program, it has many overlaps with STEP in scope and efforts are on to collaborate with each other.

Intergraph currently has development efforts that intend to support STEP standards in its CADD software in the near future (Intergraph 1997). On the implementation technology side a new standard, Object Linking and Embedding for Design and Modelling (OLE for DM) extends OLE to handle 3D objects (Wygant 1996). OLE for DM was initially developed by Intergraph and now in public domain, has been accepted by the Design and Modelling Applications Council (DMAC 1997). DMAC includes major CAD vendors and *Microsoft*. With *Windows*, OLE, OLE for D&M, users will be able to drag and drop objects across leading 3D design and modeling applications. Such a feature leads to a way of assembling a product from diverse sources as well as data exchange. It will also integrate CAD software with many current desk-top applications leading to productivity gains in report generation, etc. However this development is restricted to the *WindowsNT* and *Windows95* operating systems, for now. If OLE for DM eventually brings the same functionality to the CAD software as it has to desktop business applications, some of the trends in product data standardisation to promote integration may become function-oriented rather than the current data-oriented approach (Augenbroe 1994).

Owing to the great promise that STEP holds out for resolving integration needs of the industry many industry umbrella organisations have been formed. An example is PDES Inc. named after Product Data Exchange using STEP. It is an international consortium of companies interested in accelerated implementation of STEP (PDES 1997). There are also more focused groups such as PlantSTEP. PlantSTEP is a North American process industry initiative that is aimed at integrating STEP standards in software for use in the life-cycle of process plants such as power and petro-chemical (Edlinger 1997). Most CAD vendors and many major players in the process industry are PlantSTEP members and are committed to support the relevant STEP application protocol AP 227 for the process industry. STEP-based products are being promised by large CAD vendors. This trend will accelerate as the evolutionary nature of standard models comes to an end and the user community becomes aware of both the standards and their utility to their respective industry and begin demanding STEP-compliant products.

At AECL, the premier Canadian nuclear power research and development establishment, a 3D CADDS environment is being used in an effort to digitally assemble the next generation nuclear power plant under development. The immense complexity of such plants and previous experience with design and construction of smaller plants compels the maximum utilisation of information technology. Some of the potential benefits include:

Detection and resolution of space allocation conflicts between competing equipment and their support structures;

Provide a common reference to the different disciplines;

Visualisation and walk-through design reviews; and

Communication of the design in electronic form to client and contractors.

As an example of sub-system design issues involved in large-scale projects, an application, that the author participated in developing at AECL, is described. The support design system (SDS) uses state-of-the-art design, modeling, and computing resources to the task of integrated design of piping support structures (Ravi et al. 1997). Due to many thousands of piping and equipment support structures in a nuclear power plant, a rapid design system as well as a data management and documentation system was required. It is required to perform all design tasks from within the 3D CADD model of the nuclear plant. SDS is designed to fill this requirement by integrating 3D CADD, steel structural modelling, mechanical components modelling, stress analysis and design to nuclear code specifications. In addition, front-end integration to pipe stress analysis results via a database, back-end integration to global database of overall plant structure, interference detection with the rest of the plant design, management of stress results, 2D drawing extraction from 3D CAD model of the support, generation of bills of material and revision control are tasks SDS is targeted to support. A number of different commercial products (*PD-STRUDL, Microstation, PlantSpace, FrameWorks Plus* and ODBC database drivers) are integrated in a Visual C++ program environment to implement a client-server system architecture. SDS provides a GUI as well as facilitates the appropriate 'view' of the structural model to the analysis package by handling the data translation when required. It would be immensely advantageous, if all the different applications that are required on this software project could 'look' at the structural model in a uniform fashion to eliminate translation of engineering data from one application to another. This project, therefore, is an example where a standard model of the structure that is used by the different applications would be beneficial. It would simplify the development and use of integrated systems. A review of the current state-of-the art in information models for integrated design is presented in section 2.4.2.

2. 4 Design Models

Studies in the modelling of design were developed as a way to formalise the various facets of the design problem to enable exchange between design researchers. More importantly, such modelling strived to delineate specific design issues. Thus there emerged two views of the design - the product view and the process view. More recent developments in design modelling have to do more with the application of the computer to perform design tasks. Thus there are many models for computer-based design. Work in progress about models that borrows from developments in information processing theory and has roots in cognitive science is also contributing to the state-of-the-art. Some of the recent developments in design modelling are reviewed below.

2. 4. 1 Verbal and Symbolic Models

Traditionally the design process in the architecture and building fields has been defined as a cycle of activities including analysis, synthesis and evaluation. Holgate

(1986) describes symbolic and verbal models of the design process as follows. The verbal model is the analysis-synthesis-evaluation definition which is sometimes presented as different phases of the design process such as problem definition, postulation of solutions, evaluation of solutions and finally selection. A frequently used symbolic model is the decision tree. Symbolic models have been used extensively in the problem definition phase. Other symbolic models are the interaction matrix, interaction net, AIDA net and semi-lattice, the latter being a variation on the decision tree that reflects the fact that there could be overlap in the branching of a decision tree (Holgate 86).

More recent studies have combined research between the fields of cognitive studies and design studies and have led to new models of the design process. Oxman and Oxman (1992) state that the modelling of thought and computation as the processing of symbolic representations, the study and formalization of symbolic representations, the state description in the modelling process are all concepts from information processing theory. They describe the process models of two cognitive paradigms in design: refinement or model-based generic design, and adaptation, or case-based adaptive design. The premise is that there are two different cognitive styles at work in design. Refinement is the process in which a generic representation of a design solution is modified through formal transformations to finally result in a desired representation, the final design solution. Adaptation is the process of generating new alternative solutions by substituting some elements of the initial solution.

2. 4. 2 Information Models

The development of computer-based design systems is generally preceded by the definition of formal models of both the design objects and the design/construction process. The physical description and the semantics of the various subsystems and components are included in the product model (Evt, Khayyal and Sanvido 1992). The process model incorporates the different phases and activities of the design and construction process.

The 3P model, an information model where three axes are used to refer product, processes and the people involved in building design is presented by Bédard and Ravi (1992). It is used as the basis of a computer-based model for integration in building design. Each axis can be decomposed into several levels of detail. For example the product axis (X) can be split into the main structural, envelope, services and the interior subsystems. The model also incorporates the different steps in the process - from recognition of a need to construction, operation and maintenance of the building -along the Y-axis. The human axis (Z) includes issues from the owner, tenant, architect, engineer and associated concerns such as safety, functionality etc. The 3P model can be used as a guide to organise information in new computer-based approaches such as knowledge-based systems, object-oriented programming and database management systems to integrate and develop cross-disciplinary building design applications.

The current work has adopted the knowledge-based system (KBS) approach where there is an underlying product model as well as a process model. The product model is based on the object-oriented representation of the different building subsystems and components. It may be noted that in the context of the tool used to implement the system, the word 'frame' is used, which is synonymous with 'object'. Also, the word frame is the preferred terminology in knowledge-based systems to refer to data model objects. The tool is a hybrid KBS shell that supports object-oriented programming as well as production rules. Therefore capabilities such as hierarchical relationships between the objects, inheritance of properties between the objects and message passing are used to represent the building model (i.e. product model) as well as the design method (i.e. process model) in the system. Figure 2.1 shows a simplified conceptual model of a knowledge-based system for the design of office buildings at the initial stages of this work (Bédard and Ravi 1989) which incorporates levels of abstraction in design product and process characterisation. Björk (1989, 1994) also proposed a similar model, RATAS system, that describes building as a product, with different levels of abstraction using a hierarchy of classes and objects. The RATAS system however does not account for design process issues. The model finally proposed as the basis of Tall-D system is discussed in the next chapter.

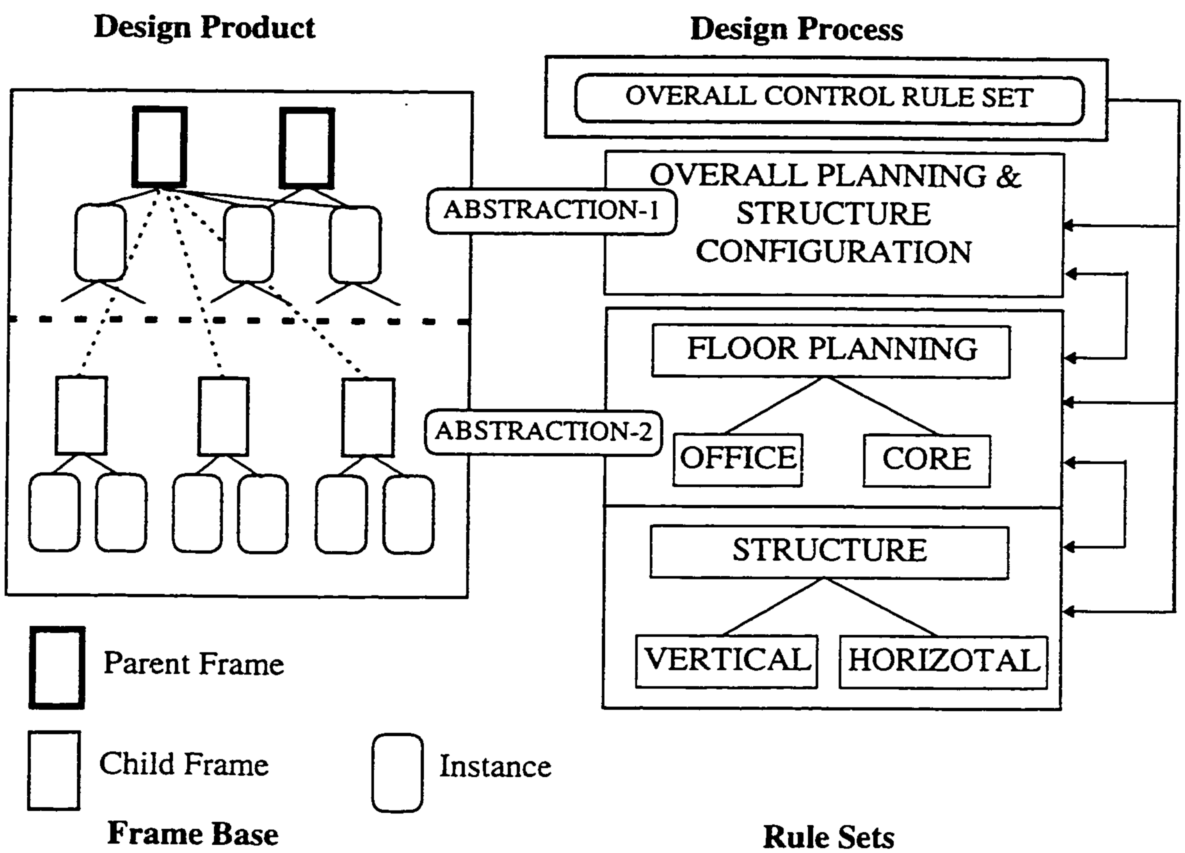


Figure 2.1. A conceptual KBS Model of the design process and design product proposed initially (Bédard and Ravi 1989).

Sause (1989) presents a discussion of the design process as related to structural engineering. In the context of computer-integrated structural engineering, an organisational model for the structural design process, termed the multilevel selection development (MSD) model, is proposed. MSD organises the process into a hierarchy of selection and development sub-problems. MSD is shown to lead to a blackboard model for structural design.

Froese (1996) reviews different projects including the STEP initiative, that are working towards use of standard data models in the areas of architecture, engineering and construction (AEC) and computer integrated construction (CIC). The prevalence of numerous approaches discussed indicates that standard models have yet to find wide-spread acceptance. However the formal release of the ISO STEP standard would move the field of standard models towards STEP and STEP-based products.

Ito(1995) discusses the pros and cons of generic product models verses specific product models. The generic model has the advantage of serving the entire design, construction and maintenance phases. However, the applications that can use the single generic model are not yet available. On the other hand specific domain or viewpoint more specifically serve an application software. Such individual models serve to advance model-based applications, ultimately facilitating the goal of generic product model for maximum possible integration.

Poon and Maher (1997) describe a nature-inspired design paradigm. It is argued that the design process is evolutionary and development of computational methods for design process can benefit from the use of genetic algorithms and evolutionary programming. Going one step further, it is claimed that design goals are not constant and hence design process has to adjust to the changing target. This process is described as co-evolutionary design. However there is need for much more work in applying genetic algorithms before practical systems like Tall-D can be built.

The process model in the present KBS centres around the design phase as opposed to the entire design/construction phase. The production rules in the system simulate the design process and constitute the active part of the knowledge-based system. A version of the generate-test paradigm is used, where the design alternatives are synthesised and evaluated (Hayes-Roth et al. 1983). The generation of design alternatives is based on the product model of the building. The design alternatives are in fact instantiations of the objects in a hierarchy of frames that describes the semantics in a building assemblage.

2. 4. 3 Means of Data Exchange

Many technologies are maturing or in final stages of development that will facilitate data exchange between applications and thereby promote integration. Two examples are object linking and embedding (OLE) for design and modeling and ISO-STEP product data definition as a standard. With OLE for design and modeling, one can move 3D design objects from one application to another without converting from one

CAD application to the other (Wygant 1996). The applications may then use the object as required by the application.

With ISO-STEP standard for the model, the intention is to use neutral files or a database to safely transfer geometry as well as design intent (ISO 1994). Therefore applications that may not communicate with each other directly can still work on the same design artifact, as STEP-compliant software should all interpret the model data uniformly. However some of the specific parts of the ISO standard are not yet finalised. PlantSTEP is an example of industry initiative to adopt the ISO/STEP standards for the process industry. CAD vendors and users are part of this consortium (Edlinger 1997).

There are many different software systems that are being used in the design office. Often the same data or quite similar data albeit in different format is required for the different application programs (Duchesne and Humar 1992). As a result, data exchange protocols have to be established to facilitate the smooth exchange of information between applications. Examples of such protocols are IGES (Initial Graphics Exchange Standard) and DXF, the latter widely used by virtue of the large dissemination of AutoCad in the industry. At best these standards are useful to transfer drawings and associated attributes between different software.

2. 5 Review of Building-Related KBS Applications

The use of KBS approach has resulted in a number of applications being developed in areas related to buildings. Below is a review of previous development work starting with building layouts, followed by buildings and structures, and finally integrated design.

2. 5. 1 Previous Work in Planning of Building Layouts

Willey (1978) presents a theoretical approach for automated architectural sketch design that encompasses the early decision making process. It is built on a hierarchical model of the design process whereby a number of potential designs are generated and pruned at each hierarchical level. Early computer-based methods of layout planning for buildings progressed from the state of automating the drafting process to developing and selecting alternative schemes (Mitchell 1977). The generation and selection processes have traditionally used algorithmic approaches. Typically, these algorithms lead to a large number of alternatives and require extensive searching before a satisfactory solution is found. Experiential knowledge is difficult to incorporate in such procedures, thus limiting their usefulness.

Significant improvements to layout and space planning automation were achieved by using more efficient generate-evaluate procedures that combine algorithmic and heuristic search methods with the help of the KBS methodology. Liggett (1990) has developed a computerised approach in which the two traditional methods of algorithmic generation of alternatives and manual construction of the layout by means of a graphical interface are used to solve space allocation problems. By combining these two methods, it is possible to overcome the limitations specific to each one, i.e. of the first method that is not flexible enough since it excludes interaction with the designer, and of the second method that denies access to computer-based decision aids in generating and evaluating design options. Space allocation for activities in new or existing buildings is then performed, with a graphical interface to help in the layout process and evaluation of solutions. Another system for developing building layouts (Flemming et al. 1988) complements the abilities of the human designer in the enumeration of alternatives and the consideration of diverse criteria. A domain-independent generator for the layout of rectangular modules is used in conjunction with a domain-specific tester in such a way that the system can be applied to more than one domain. Layout of kitchens and layout of multistorey building service cores were performed with the assistance of this system. Automated space planning is also performed by a system called HeGeL (Akin et al. 1988) in which the heuristic generation of floor plan layouts is based on a formal model of the architectural design process. Given an outline, HeGeL generates alternative locations for design units, say a furniture set, accounting for constraints like direct access, natural light, privacy etc.

2. 5. 2 Applications in Structural and Building Engineering

HI-RISE is a preliminary design system for assistance in the design of commercial and residential high rise buildings (Maher and Fenves 1984). A three dimensional grid of spatial planning is given as input and remains fixed through the design cycle. The system generates alternative structural configurations, by synthesizing generic subsystems defined in the knowledge-base. Only one alternative subsystem at a level is permitted, thereby limiting the choice of alternatives that could otherwise be developed.

A conceptual model, DESTINY, was proposed as a framework encompassing all phases of structural design (Sriram 1986). The model suggested the use of a blackboard architecture to manage different knowledge modules. ALL-RISE, a partial implementation as a preliminary design system developed for assisting in all types of buildings, is part of the DESTINY model. ALL-RISE was developed as an extension of HI-RISE to include low and medium rise buildings. ALL-RISE was developed as a case study in constraint-based design. It generates many structural alternatives based on the input of spatial information after architectural planning. However the alternatives generated and evaluated result in only a partial solution, i.e. only location alternatives are generated. Structural components are not generated and structural response of the system is not considered. The latter two steps represent design operations at the subsequent conceptual level of detail of the problem solving strategy employed in ALL-RISE.

Fenves et al. (1995) describe the conceptual structural design system SEEDConfig that is part of the SEED project. SEED stands for the Software Environment to Support Early Phases in Building Design and is reviewed in section 2.5.3. A previously generated geometric model by the massing submodule of SEED is used by SEEDConfig as the starting point. Potentially applicable solutions are stored as a technology tree that is claimed as an extension of the concept of a toolkit of potential solutions. In combination with case-based reasoning, the system is designed to retrieve previous designs for use in a similar design context. The SEEDConfig system has five components - input, problem specification, generation, evaluation and output modules. The generation module uses different levels of detail or abstraction. It generates alternative feasible solutions in a hierarchy of solutions in tandem with the similarly arranged technology tree. This feature makes it similar to the hierarchical generate test feature used in Tall-D to generate the alternative spatial configurations of tall buildings. It is necessary to backtrack in SEEDConfig to generate a different alternative whereas Tall-D can simultaneously generate and reason with multiple alternatives at the different levels of abstraction. Tall-D generates the initial geometry and does not need a 3D geometry input to begin preliminary design. The structural evaluation module in SEEDConfig is under development and mostly functional criteria (i.e. load paths, lateral load resistance etc.) are envisaged but not any relative comparison of the alternatives.

The Building Envelope Analysis and Design System (BEADS) completed at the Centre for Building Studies is a KBS to assist designers at the preliminary stage of envelope design (Gowri 1990). BEADS helps designers to consider a large number of alternative material and construction systems for the envelope. Generation of feasible alternatives is performed as a constraint-based search problem, where components of the building envelope must satisfy performance requirements. Problem decomposition and multiple levels of generate and test paradigms are used. Ranking and selection from the alternatives is done after evaluation against an array of performance criteria as well as priorities supplied by the designer.

Harty and Danaher (1997) describe DOLMEN, a knowledge-based system that is primarily targeted to help designers evaluate preliminary structural designs. To evaluate designs it needs input of structural systems details (for each alternative). Then, DOLMEN uses traditional utility theory of decision analysis to arrive at a numerical value for the given structural design in up to eleven evaluation criteria. However criteria used in Tall-D for evaluation of overall building configurations such as service connections, windowline, potential for good structural design and energy efficiency are not present in DOLMEN. DOLMEN needs input of structural geometry such as bay sizes, number of storeys and floor height, which aspect is similar to HI-RISE. Tall-D on the other hand begins design at a more basic conceptual level, considering the aforesaid parameters variable and proceeding to finer structural details.

Hauser and Scherer (1997) present a system for the preliminary design of reinforced concrete structures of industrial buildings. It uses heuristic search, constraint programming and planning methods to apply intelligent CAD paradigms for structural design. The system implements a theory of abstraction that distinguishes between many different design decisions. It is argued that design models and corresponding design reasoning can be expressed in hierarchical levels. This idea is similar to the abstraction levels and hierarchical generate-test paradigm used in the current work on Tall-D. The system described by the authors does not generate alternative structural schemes for a given design problem as in Tall-D, but generates a solution interacting with the user in a CAD-like environment. In that sense it is a generic concrete design system rather than a preliminary design system in the mould of Tall-D.

Using a Structural Design Language (SDL) a KBS for the design of moment-resisting steel frames was developed (Paek and Adeli 1988). There are four modules in the system for connection design, section design, frame analysis and frame predesign. The stiffness matrix method is used as in conventional frame analysis programmes. The frame predesign module selects initial sections from a set of predefined choices derived from literature.

A KBS developed by Jayachandran and Tsapatsaris (1988) is restricted to the selection of the lateral load resisting system. No preliminary structural design is carried out. The system thus helps only in the selection of the type of structure rather than performing alternative structural designs.

CONCEPTUAL is another system which helps in the conceptual design of structural systems (Haber and Karshenas 1990). The selection of the structural type is made from a predefined database of components. The systems emphasis is on the cost estimates for different building alternatives.

INDEX is a KBS for the detailed design of industrial structures. Preliminary analysis routines are implemented in PROLOG and detailed analysis in FORTRAN-77 (Kumar and Topping 1988). Preliminary analysis and design of steel frames such as portal frames, roof trusses as well as beams and columns can be performed. Detailed analysis is performed through an interface to the PROLOG system.

In another development, conceptual design of office steel buildings is performed by a system based on first order predicate logic (Jain, Krawinkler and Law 1991). Preliminary structural design is attempted by a new approach of using first principles, implemented by symbolic logic, rather than heuristics. The system starts with the generation of gravity floor framing followed by the lateral load resisting structure. Only rigid frames could be considered for the latter. The main advantages of the approach claimed are the expressive power, inferential capabilities, and cleanliness and elegance of expression resulting in knowledge transparency. However many researchers have expressed reservations as to the capability of logic systems to model human problem solving. Logic systems could also be cumbersome and much slower.

Turk et al. (1994) describe an European research project that tackles the bane of many engineering firms. Long proven and existing structural analysis and design programs that do not readily fit into current computing technology but cannot be given up due to their being knowledge repositories accumulated over time. This project aims to integrate and support such popular applications by implementing an underlying object oriented data model of the building/structure. Called the Dimensioning and Analysis of Reinforced Concrete Structures (DIANAS) project it loosely couples programs developed in languages such as Fortran for the analysis and design of structures. The different coupled programs performed tasks such as architectural geometry, structural idealization (or model) and two mathematical models. Translation from one task to another was originally managed by the user with data files which is now handled by DIANAS. The integration in the analysis and design process is being facilitated by a STEP aligned object oriented model of the facility.

Case-based preliminary design in a system known as CADRE is discussed by Bailey and Smith (1994). The system uses cases as a starting point for further adaptation rather than routine indexing and browsing of cases. Architectural and structural domains are sought to be integrated by using a case memory of existing designs for which integration issues have been addressed. The custom adaptation of a case is primarily done by solving a set of linear and non-linear constraints describing the architectural and structural abstractions of a building. If this does not produce a design solution, topological adaptation is attempted which yields a new set of geometric constraints. The concept of the system is based on the belief that only one solution need be pursued. However, preliminary design systems have a superior performance from the users point of view when alternative solutions are considered without precluding all but one as CADRE does. This aspect of multiple alternatives at the preliminary stage is appreciated by designers as is shown in the questionnaire answered by designers evaluating the Tall-D system (see section 6.3.7).

Maher and Garza (1996) discuss the application of case-based reasoning (CBR) for structural design. Experienced engineers at a design firm along with some of their designs, i.e. drawings and specifications, formed the basis to the development of case-base. To represent case memory, attribute-value pairs are used and arranged into sub-systems and form hierarchies in case memory organisation and indexing. The attribute-value pairs are also grouped into functional, behavioral and structural roles. Such attribute categorisation is said to provide an effective way of reasoning about various design variables. In addition to specific cases in case memory, the system CaseCAD also has a knowledge-base of generic types of designs in the form of predefined models, for example a medium rise building's characteristics. The user begins with specifying some broad criteria and CaseCAD allows the user to browse graphically through the relevant cases retrieved. The user can select a case for adaptation if he sees it meets his important criteria. Tall-D generates alternative building structural designs based on the context information and most of them meet the criteria set out initially as well as allowing the user to proceed with a particular type of system. CaseCAD, however, cannot adapt the retrieved case, which the user has to do using object editing tools, *AutoCAD* and *XFIG*

2D drawing package. The new case can then be added to the case memory. Thus CBR, unless fully developed with adequate and automatic case adaptation function, may serve as little more than case browsers leaving a substantial portion of the design to be completed by the designer. The designer is therefore left without many relevant, complete alternatives to evaluate. Tall-D however presents many relevant and complete structural design alternatives for the designer to evaluate.

Enseleit et al. (1995) discuss the use of STEP standards and tools for distributed structural analysis. Using finite element analysis (FEA) as an example the study identified the difficulty in consistently transferring data in STEP format to a format suitable for the analysis package. The STEP format is also stated to be less efficient for large-scale FEA analysis. The work also found STEP lacking in features for communication across a network as well as secure access and version management of product model data.

Turkiyyah and Fenves (1996) discuss a knowledge-based assistant for finite element modelling for structural analysis. The often meticulous task of finite element discretisation requires an expert. The results are best interpreted by experts who are aware of the underlying assumptions of the model. This work is an effort to develop a knowledge-based assistant that can present the user with high level abstractions of the underlying model at the same time be closely coupled with the finite element solver. The high level abstractions allow the user to interact with the system in terms of problem features, modelling assumptions and performance criteria. The approach uses various geometric, functional, and behavioral abstractions as well as a hierarchical modelling strategy that progressively relaxes the constraints on an initial crude model.

2. 5. 3 Previous Work in Integrated Design Systems

Several research efforts can be reported about integrated design. An Integrated Building Design Environment called IBDE (Fenves et al. 1990) is a large exploratory system that can account for the vertical integration of design activities from conceptual floor layout to construction planning. It is based on a blackboard architecture and used as a test-bed for ideas and concepts regarding the development of integrated systems in engineering design. IBDE currently incorporates seven distinct KBS, each dealing with a specific task in the design and construction planning of multistorey office buildings. It starts with a submodule ARCHPLAN capable of generating schematic building designs with a limited number of core configurations and finishes with CONSTRUCTION PLANEX to provide a schedule of construction activities with cost estimates.

Flemming and Woodbury (1995) present a overview of the Software Environment for Early Phases in Building Design (SEED) project. SEED integrates many distinct modules for preliminary building design. SEED uses many modules that were part of the IBDE described above. The different modules are SEED-Pro, SEED-Layout and SEED-Config. SEED-Pro supports the generation of an architectural plan in terms of the basic functional units. SEED-Layout supports the schematic layout of the previously generated functional units. And finally SEED-Config generates a three dimensional geometric model and then alternative structural systems for that geometry. SEED is unique in its effort to provide a systematic framework of capturing design experience out of an organisations recurring building types and use it for generating designs that are similar. Case-based design is the main paradigm used to implement SEED. The structural design module in SEED is dependent on the richness of the case-base it begins with. Tall-D on the other hand assembles structural design alternatives based on the context without the need for refinement of nor the limitations of predefined solutions. While the case-base of SEED modules are enhanced by adding to the case-base, Tall-D knowledge-base of production rules and objects need similar maintenance to enhance or refine the solution set.

A computer-based system called DICE (Distributed and Integrated environment for Computer-aided Engineering), aimed at addressing the coordination and communication problem in engineering, is described by Sriram et al. (1990). If the design concept and intent is conveyed cumulatively from preliminary design through fabrication and construction, and the key players at the different phases are forced to consider the whole picture, then many design errors can be avoided. The purpose of DICE is to integrate a suite of applications that will help large engineering projects to achieve that level of coordination and communication which leads to a better design and design process. DICE is designed for a network of computers and users. An intelligent global database acts as a blackboard system for DICE and is implemented using a commercial object-oriented database system GEMSTONE. The rest of the system consists of many knowledge-modules representing many domains. The knowledge-modules may be any of KBES, a CAD tool, an analysis tool or even a user. The knowledge-modules are coordinated by the Coordination module of the blackboard. Blackboard consists two other parts: Solution and Negotiation. The Solution part contains the solutions generated by the knowledge modules. The Negotiation part keeps a trace of the negotiation between the different engineers taking part in the cooperative design development. The system seems more suitable for detailed design as the accumulation of design rationale across phases and disciplines is maximised as the design proceeds through detailed design and fabrication/erection/construction. The detailed design knowledge-modules were not fully developed and, further more its scope is being restricted to steel structures. This system is aimed at assisting large engineering projects to avoid serious design errors. Large projects will be in a position to take advantage of this resource intensive process. Smaller projects, however, may need more modest and but effective means of achieving the same degree of design safety.

Intelligent Integrated Building Design Systems (IIBDS) represents another effort at integration in the European Community countries (Augenbroe 1994). As a first effort, the pilot project COMBINE (Computer Models for the Building Industry in Europe) was initiated in 1990. It set out to study a multi-disciplinary approach from the design viewpoint and also to integrate existing tools in the building process. All relevant issues in the energy, service, functional and performance characteristics were identified so that these characteristics could be oriented to the same computer modelling process. The project evolved to COMBINE 2, where a data model, the Integrated Data Model (IDM), is used as a basis to facilitate different applications tools to work with a common building model (COMBINE 1997). IDM was further expanded with Data Exchange System (DES) that allows architects and engineers to communicate and exchange design information though a common format. IIBDS is to be made available with a choice of two commercial

CAD applications so that current CAD users will be in a position to take advantage of the system.

2. 6 Summary

With the exception of IBDE and its sequel the SEED project, the systems reviewed in sections 2.5.1 through 2.5.3 generate designs at the detailed level, not at the overall level of buildings. The KBS presented here is aimed at the overall configuration of multistorey office buildings. The main tasks being to generate building configurations with the required rentable area and to develop an efficient structural system within the constraints imposed by the owner and designer. It does not address the spatial arrangement or subdivision of functional zones within the rentable area. While the previous systems consider layout planning primarily from an architectural perspective, the present system also takes into account engineering constraints in configuring multistorey office buildings. The current system goes beyond the location of the different lateral load-resisting structural systems, as in ALL-RISE, to generate the structural components and evaluate the structural response. Structural systems are generated, based on the design context, as opposed to the use of generic alternatives as in HI-RISE predefined in the knowledge-base.

This chapter, in part has, presented a review of the state-of-the art in integrated design especially applied to buildings and other constructed facilities. In another part, the state-of-the-art in knowledge-based systems application to integrated design was presented. The different modelling perspectives were presented. Design process modelling and product modelling were discussed in relation to the current work. The early state of data exchange between applications and its current state, i.e. the development of standards to facilitate the exchange of design intent in addition to data, were also reviewed. The state-of-the-art shows that there are further research efforts required before integrated design of the engineering of artifacts can be achieved.

CHAPTER 3

Overview of the Tall-D System and Implementation Considerations

In this chapter the overall strategy used in the research project to develop the Tall-D preliminary design system is described. The knowledge-based model used to implement the system is discussed. The functions of the different components are introduced. System implementation issues as well as sources and types of information used are also discussed. Terms used in this chapter that are related to the knowledge-based system (KBS) approach are defined in a glossary in Appendix A-4. In order to discuss implementation issues a typing convention is used in this chapter. *Italics* typeface is used for frames used in knowledge representation while `courier` font typeface is used for instances of frames as well as Lisp routines, variables and function names. Italics may in places be used for sub-titles in an obvious manner not to conflict with the above convention. It may also be noted that the words frame and object are used interchangeably in the context of Lisp-based object oriented programming.

3. 1 Overall Strategy

The overall strategy followed in the Tall-D system development is to maintain a close similarity to the design process in practice, yet take advantage of the computer to perform rapid generation and evaluation of alternatives. Thus it can be seen that the generate-test paradigm is a suitable method of modelling the design process. However to manage interactions between different modules, agenda control is employed. Agenda control is performed by the inference engine using higher level rules (or macro rules). These macro rules incorporate the knowledge about the design process to activate or deactivate a knowledge module. For example, when a particular command is issued by the designer during a design session, the macro rules determine the appropriate knowledge module(s) to be used, which is(are) then activated for inclusion in the inferencing process.

A knowledge module consists of sponsors arranged in a hierarchy, where each sponsor consists of rule-sets grouped to perform a specific design task. Agenda control essentially works by allocating system resources to the desired sponsors. Such priority given to sponsors and rule-sets enables the system to focus on the appropriate group of rules and decisions as the design progresses.

Frames are used to represent building configurations at multiple levels of details (or abstractions). By applying the generate-test paradigm at different levels of detail, it is easier to focus on the primary design parameters under consideration at each level. The major advantage of this approach is the possibility of considering a large set of alternatives at the initial stages of the generation process while performing design checks to narrow down the solution space to a smaller but more detailed and relevant set of alternatives, with attendant benefits for the designer. This precludes the combinatorial explosion of alternatives as well as facilitates the development of details, thereby reducing the overall extent of search for a solution.

Figure 3.1 is a schematic representation showing the major components of the building design KBS. The user interface is a graphical environment that allows the interactive participation of the designer in performing preliminary design. Graphical displays of structural system alternatives as well as floor plans of different building configurations are possible. The components of the knowledge-base represent the aforementioned two domains of structural engineering and architecture. The outer circular region with sub-divisions in Fig. 3.1 is representative of the way the knowledge-base is segmented to support different knowledge representations such as rule-sets, frames etc. The inner-most region groups meta-rules to control the course of design tasks. These groups of rules put generic tasks to be performed on the agenda of the inference engine. Such tasks serve to

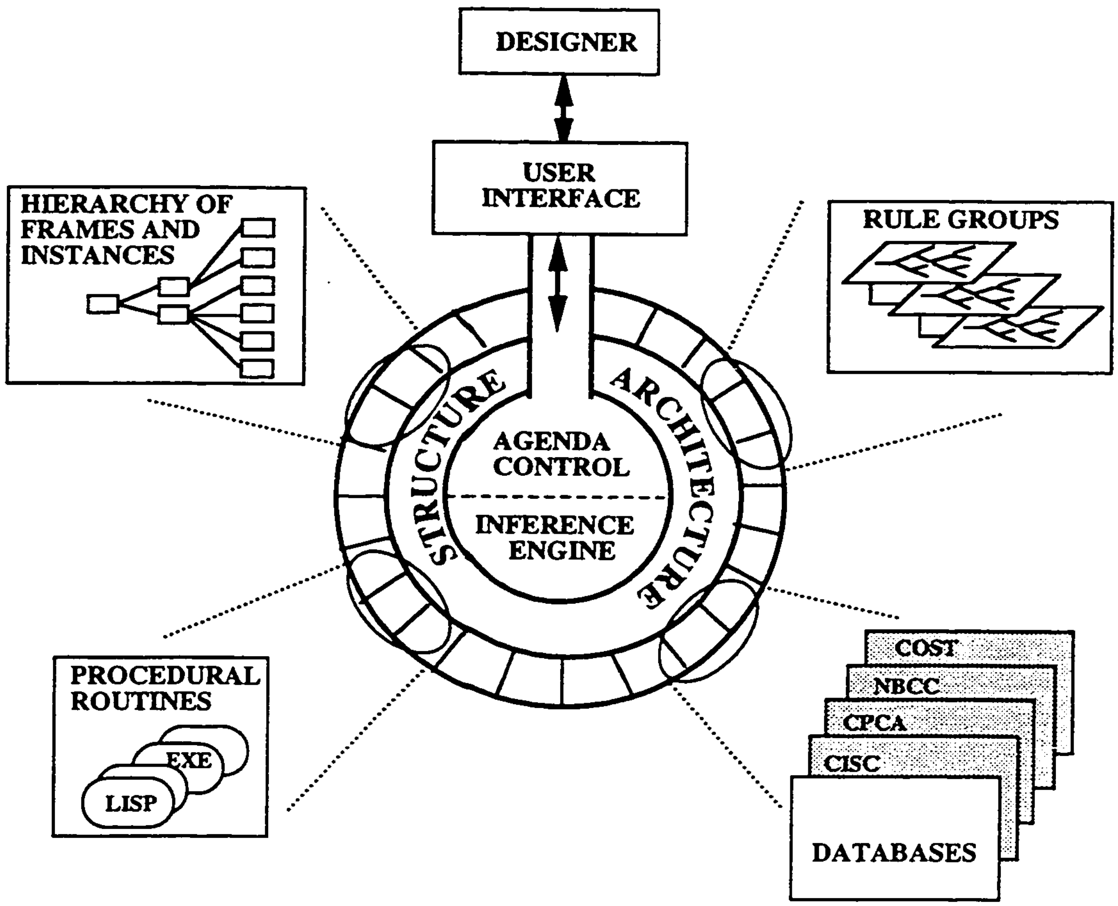


Figure 3.1 Schematic representation of the Tall-D building design simulation KBS.

initiate the inferencing process by transferring control to an appropriate rule-set responsible to carry out a specific design task.

The inference engine which is capable of both forward chaining, backward chaining and mixed chaining, is primarily used in the forward chaining mode. The reason is that design is primarily a data driven process, for which forward chaining is most suited. The inference engine in forward chaining checks rules that have their IF part (antecedent) matched by assertions in the current state of the knowledge-base and selects to fire or trigger their THEN part (consequent). The rules as mentioned above are grouped under rule-sets and in turn assigned to sponsors. Each rule-set consists of rules pertaining to a specific design task to be carried out. The macro rules in the system activate the desired sponsor that will enable the use of its rule-sets to perform the next design step. Sometimes such rule-set activation is done by Lisp routines associated with the menu items in the Tall-D user interface. Thus the inference engine uses the different knowledge modules to complete the design cycle.

The design product or the building is represented in the knowledge-base as a semantic network of frames. Instances of frames are generated and context-specific values are assigned to attributes (or slots). The heuristics employed in the building design process are modelled by the rules. Rules from different rule-sets operate with the initially available information in knowledge-base. During consultation, design information is added in newly generated instances of relevant frames.

Procedural functions are necessary to carry out algorithmic/numerical computations in engineering design application. Procedural programs that are invoked by the rules to perform numerical calculations are grouped as a separate entity of Lisp routines. Data such as the CISC steel tables are stored as external data files. However CISC steel tables interface was not used in Tall-D eventually, for reasons explained in section 3.5.2. Knowledge representation is described in more detail in sections 3.3 and 3.4.

3. 2 Programming Issues

As explained briefly in the previous section, a hybrid knowledge representation, supporting both frames and production rules, is used in the implementation of the system. Such a KBS environment is more appropriate than a simple rule-based system for the development of design KBS. Building design deals with physical artifacts and their associated attributes such as geometry, functionality and performance. These artifacts are conveniently modelled as frames. A comprehensive representation of artifacts' attributes is achieved by virtue of the semantic relationships between the frames. The use of frames also reduces the number of rules that would otherwise be required to define the basic facts and semantics of the artifact in addition to the design process. Such a hybrid mode of knowledge representation also facilitates the separation in the knowledge-base between the "what" and the "how", the latter corresponding to inferential knowledge and is modelled by rules. In the following, a brief description of the salient features of the KBS development tool is presented which shows the suitability of the tool chosen.

3. 2. 1 KBS Development Tool

The KBS development tool GOLDWORKS-II and GCLISP Developer (GoldHill 1989, GoldHill 1997) were used in the implementation of the system. Building layout generation and structural configuration involve the representation and use of diverse types of information, parameters and constraints (geometric and other). The shell provides frames for representing design objects such as building configurations, and production rules to represent inferential knowledge.

To describe design objects and their attributes, frames are used as templates for creating a generic knowledge representation scheme. Instances are generated as context-specific replicas inheriting the characteristics of the parent frame, with specific values in the slots representing attributes. Layout alternatives for example are modelled as instances with attribute values assigned to the slots. Constraints and demon functions can also be associated with slots. The constraints may be numerical or symbolic and define the valid range or valid choice of slot values. Demon functions pertain to slots, in that they are invoked when the value in the slot is modified or even accessed (based on a switch). Such functions are defined in the underlying language of the KBS development environment - Lisp in the case of GoldWorks. Thus these functions can effect changes in other instances, create new assertions in the knowledge-base, or even carry out numerical calculations based on the change in the slot value.

A handler in GoldWorks is associated with frames and inherited by all their instances. It is also known as a member function of the object, defined in Lisp. A handler can be invoked only by sending a message to the appropriate instance. Handlers, thus facilitate message-passing between instances, and also one-way messages from within rules as well as from demon functions associated with slots of instances. The handler has ready access to all the slot-values of the instance. They are used to effect changes as required in the knowledge-base. Upon receipt of a message the instance's handler typically performs calculations using the slot values.

The GoldWorks tool is based on the Common Lisp Object System (CLOS) which is an object oriented extension of Lisp. It therefore is able to provide a good support for any object oriented modelling in developing KBS applications. The use of objects or frames as they are known in Lisp systems in knowledge representation is complemented by production rules to model inferential knowledge, thus creating a mixed paradigm often referred to as hybrid knowledge representation. The rules use the information in different instances representing owner requirements, Code constraints and design context, to generate configuration alternatives at different levels as design proceeds. Rules are assigned to sponsors, i.e. entities used to allocate priority to sets of rules aimed at a sub-task in the design process. Such sponsors are arranged in a hierarchy and controlled by macro rules that activate/deactivate sponsors. This provides the required control mechanism to realise the KBS.

An interface is provided by the development tool to help specify frames and rules. A Lisp programmer development environment (the GCLISP Developer) can also be used for this purpose as well as in cases where maximum flexibility and control is required in defining the rules. Additional components of the system like demon functions and handler functions (or member functions) may only be defined with the programmers interface. Procedural functions used for structural analysis and sizing purposes are defined using the above facility. Appendix C (see page C-28) lists the names of more than 75 structural design Lisp routines in Tall-D. In fact names of all KBS components (frames, rule-sets, rules, sponsors, functions etc.) used in Tall-D are found in different sections of Appendix C. See Table 3.1 for a summary of different KBS components used in the Tall-D system.

The user interface consists of a graphic display that helps the designer to specify design and evaluation preferences, and more importantly to keep the designer posted with graphical representation of the design alternatives being generated. The floor plans and structural system alternatives are displayed using the graphical interface. In the development of the system the user interface is considered an integral part along with the design object. Fig. 3.3 and 3.4 described in section 3.3 show how building objects inherit attributes from graphic objects, among others. Thus, an important part in a KBS for design that is often overlooked - the user interface - is developed into an effective front end, as will be seen from screens showing plans and elevations of building designs presented in Chapter 5. A ready check of the physical representation of the design decisions is thus possible and represents a close approximation of the way designers work with sketches and drawings. It therefore enhances the system as a building design tool by facilitating visualisation of the emerging design alternatives. The names of the Lisp routines performing some of the graphical user interface function are listed in Appendix C (see pages C-33 to C-39).

3. 3 Knowledge Representation of the Building

This section describes the object-oriented features used in the Tall-D systems knowledge representation. Table 3.1 shows the Tall-D system metrics - the number of rules, frames and other items in the knowledge base. The names of all these rules, frames and functions are listed in sections C-1 to C-7 of Appendix C. The names in most part are descriptive of their purpose or function. Contents of these sections of Appendix C though not discussed considerably, will make a good complement to the Tall-D implementation aspects discussed in this chapter by providing a discernable outline of the entire program. A complete listing of the program code was too long to present, running into more than 500 pages. Hence the brief overview in Appendix C.

Table 3.1 Approximate system metrics of the Tall-D preliminary design system.

KBS Components	Number of occurrences in Tall-D
Sponsors	9
Rule-sets	25
If-Then Rules	415
Frames	110
Lisp Functions	250
Run-Time Instances of Frames	200 to 400

The more than 400 rules are grouped under rules-sets. The names of different rule-sets and their primary function are shown in Table 3.2 which provides an overview of the rule base in Tall-D.

The more than 100 frames in the Tall-D system make a library of objects related to building design. See Appendix C, page C-22 for a list of these frames. This object library is used to model the physical building. Following is an illustration of this with examples from Tall-D that show the building configuration and structural system components in a semantic network.

3. 3. 1 Representation of Building and Structural Schemes

Knowledge representation of the building configuration and structural schemes is illustrated using Figures 3.2, 3.3 and 3.4. Figure 3.2 shows in a 3D illustration an example building generated by Tall-D. The entire wireframe diagram can be said to be a partial view of the internal (to knowledge base) *Building-Configuration* object which encapsulates the structural and architectural aspects of the building. A partial view, because what is seen is the geometric view and the relative position in space of the different objects. A building configuration alterative is formed when the designer selects a particular floor layout to proceed with. Each building configuration alternative therefore is one in which the outside geometry and the core geometry is fixed - ie. the massing is fixed. It is then possible to generate many structural system alternatives for this particular geometry incorporated by the building configuration alternative. Tall-D therefore generates the different structural system alternatives suitable for the building configuration. The wireframe drawings in Fig.3.2 are also an illustration of a data interface in Tall-D, that facilitates export of the building geometry to third party CAD programs, which is described in section 3.5.1.

Object Hierarchy for Lateral Load-Resisting Systems

Lateral structural systems can take different geometric forms, and can benefit from use of a sophisticated object-based knowledge representation. Use of a grid in some cases

Table 3.2 Division of rules in Tall-D system as Rule-Sets focused on a task.

Task or Area of Rules	*Name of Rule-Set*
Architectural Rule-Sets	Generate-floor-Rules Evaluate-Floor-Rules Geometric-Information-Rules
Cost Rule-Set	Approximate-Cost-Rules
Generic Structural System Rule-Sets	Construction-Material-Rules Structure-Type-Selection-Rules Activate-Structure-SubModule-Rules
Lateral Structural System (LLS) Rule-Sets	Semi-Rigid-Frame-Configuration-Rules Steel-Rigid-Frame-Configuration-Rules Braced-Frame-Configuration-Rules Concrete-Rigid-Frame-Configuration-Rules Frame-ShearWall-Configuration-Rules Framed-Tube-Concrete-Configuration-Rules Framed-Tube-Steel-Configuration-Rules
Gravity Structural System (GLS) Rule-Sets	Generate-Gravity-System-Options-Rules Concrete-Gravity-System-Rules Concrete-Gravity-Scaling-Rules Steel-Gravity-System-Rules Steel-Gravity-Scaling-Rules

is sufficient to represent simple column layouts. However a grid is not adequate to represent many practical structural arrangements. A grid also does not lend itself well to formalism in order to write production rules in a hybrid knowledge representation environment. In previous work related to this, a structural grid is most commonly used (Maher 1984, Sriram 1986, Jayachandran and Tsapastaris 1988, Karshenas 1992). However in Tall-D a comprehensive set of objects are used with names ending with *scheme* and inheriting from *Column-Layout* and *Super-structure* objects to represent the various structural schemes. Table 3.3 shows a list of such structural scheme objects in the Tall-D system.

Table 3.3 List of Structural Scheme Objects in Tall-D knowledge base.

Structural Scheme Object in Tall-D	*Primary Construction Material*
Concrete-Rigid-Frame-Scheme	Concrete
Steel-Rigid-Frame-Scheme	Steel
Braced-Frame-Scheme	Steel
Shear-Wall-Scheme	Concrete
Frame-ShearWall-Scheme	Concrete and Steel
Framed-Tube-Scheme	Concrete and Steel
Semi-rigid-scheme-perimeter-frame	Steel
Semi-rigid-scheme-Int-Braced-frame	Steel
Semi-rigid-scheme-Ext-Braced-frame	Steel

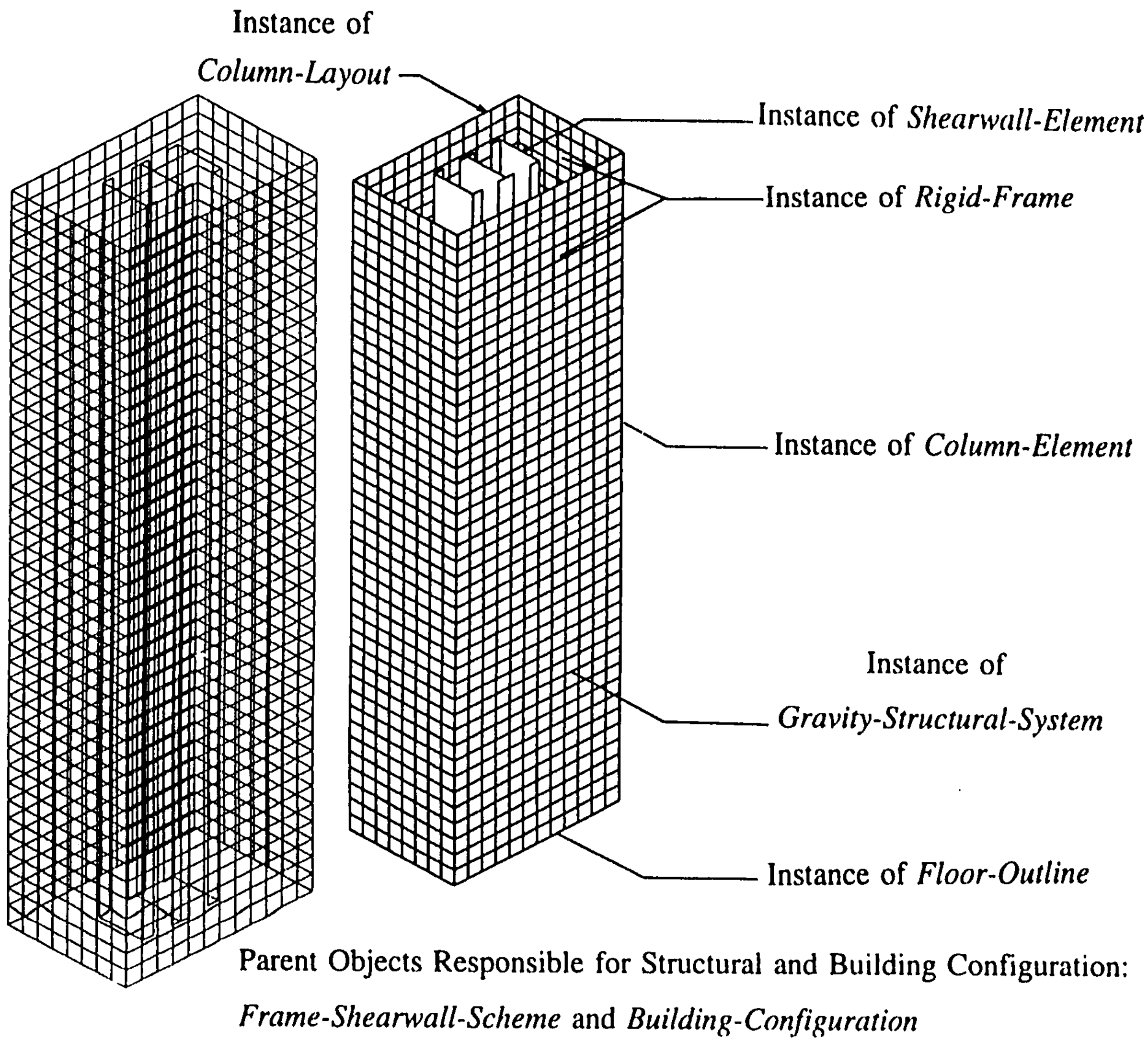


Figure 3.2 Instances of Tall-D knowledge-base objects representing multistorey building configuration.

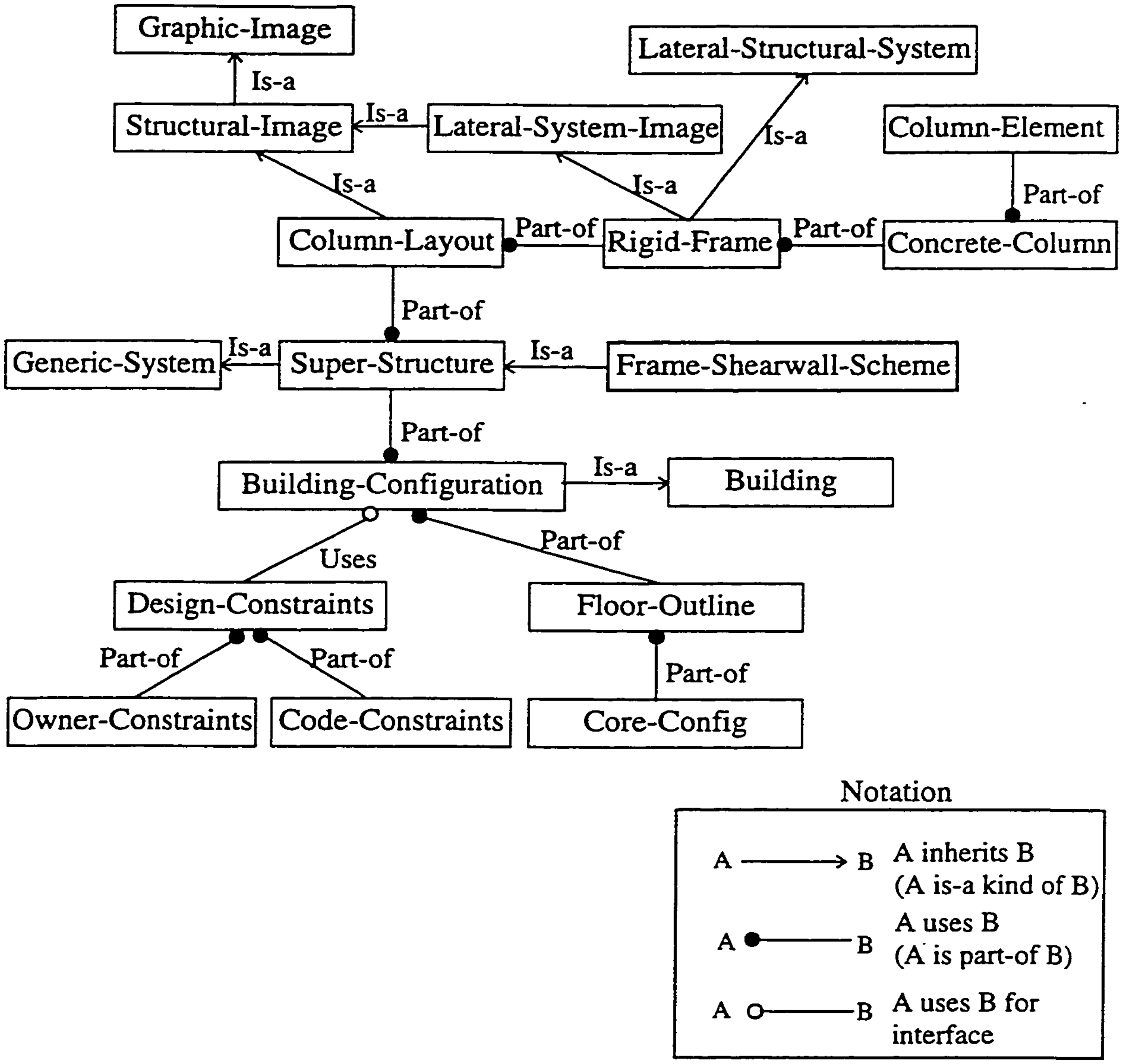


Figure 3.3 A partial object hierarchy in knowledge base illustrating the semantics between objects closely related to LLS configuration for frame shearwall interaction structures.

Figure 3.3 shows a partial semantic network of objects that together model the lateral load-resisting system (LLS) of the building. Here, only the objects related to frame-shearwall type of structures are retained for the sake of clarity. Other such structural schemes that could be represented in a similar semantic network are those in Table 3.3, many of which may be suitable for one given building configuration. As a result, instances if all those suitable schemes may be present simultaneously in the knowledge-base. The concept of structural schemes and the development of the geometry for the schemes are discussed in sections 5.1.1 and 5.1.2 respectively.

Object Hierarchy for Gravity Load-Resisting Systems

Approximately one-third of the rules in the Tall-D knowledge-base represent the gravity system configuration. A major part of the gravity system configuration is the identification of the different spans and locating the controlling span. The controlling span is the one that determines the total depth of the gravity system due to the maximum depth of the beam being in that span. Since such controlling span depends closely on the column layout, GLS configuration is linked to the *Column-Layout* object which in-turn is closely related to LLS configuration. Thus the gravity system is influenced by and influences the LLS.

Figure 3.4 shows a partial semantic network of objects that together model the gravity load-resisting system (GLS) of the building. Here, only the objects related to one-way beam-slab floor systems (shown in boxes with dotted lines) are retained for the sake of clarity. Other such gravity systems that could be represented in a similar semantic network (by filling the dotted boxes with appropriate objects) and supported in Tall-D are discussed elsewhere in section 5.2.1 and Table 5.1. Many of these GLS may be suitable for a given building configuration. As a result instances of all those suitable GLS alternatives may be present simultaneously in the knowledge-base. The issues related to GLS are discussed in detail in sections under 5.2 of chapter 5. Comparing Figs. 3.3 and 3.4 it can be seen that there are a lot of common objects between the LLS, the GLS and the underlying architectural subsystems. This aspect constitutes part of the domain integration from the point of view of product data modelling. The other part of integration is achieved in developing the rule base in conjunction with the physical product description that these semantically related objects represent, and incorporating the design process from the point of view of the architectural and structural domains.

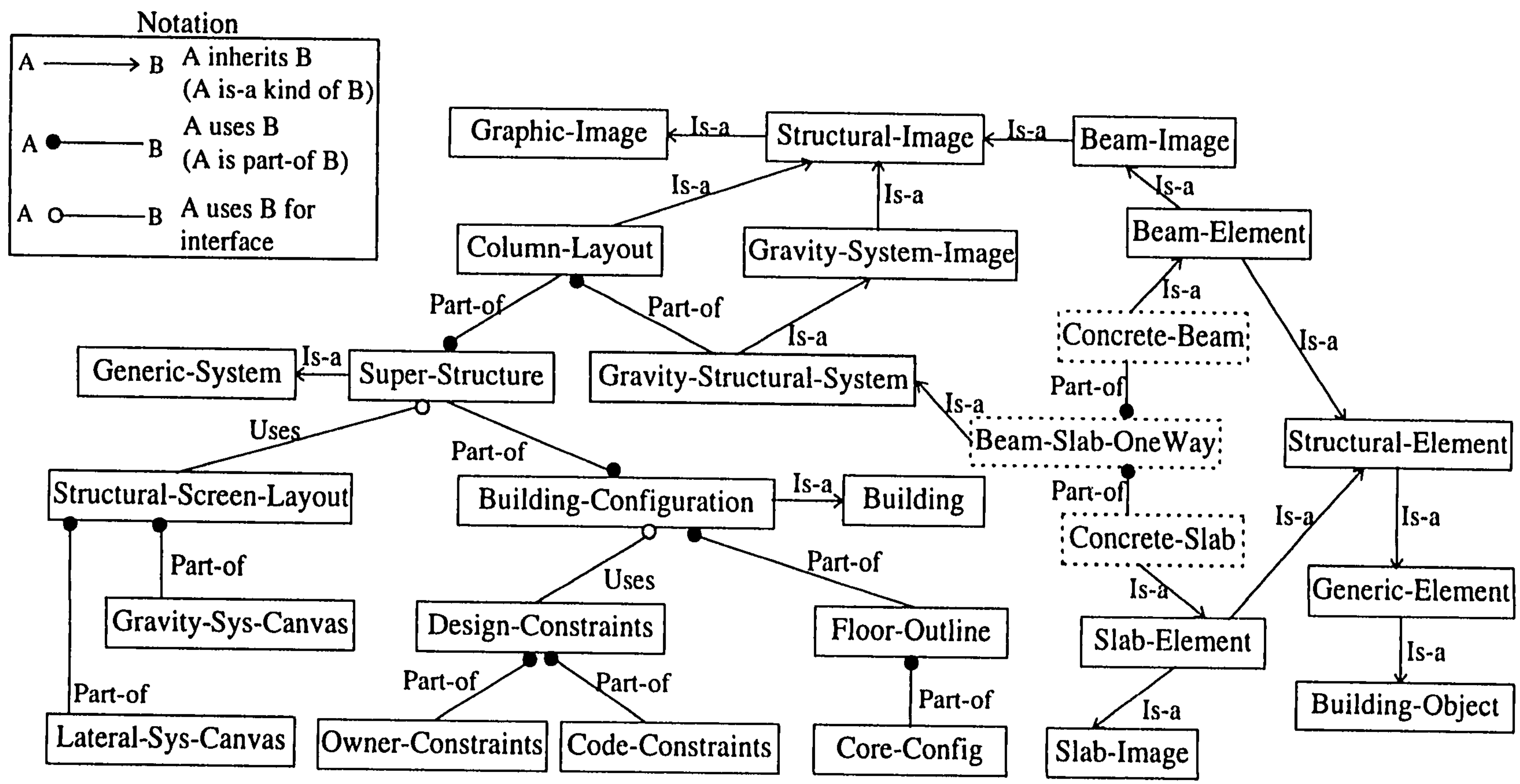


Figure 3.4 A partial object hierarchy in Tall-D illustrating the semantics between objects closely related to GLS configuration.

3. 3. 2 Representation of Structural Members and Steel Sections

The representation of individual structural elements such as columns, beams and slabs is governed to some extent by the limitations of the computer in terms of memory. In this context, it may be noted that the version of the KBS tool used (GoldWorksII) could only use a maximum of sixteen mega-bytes of memory (RAM). While using an instance of an object to represent every occurrence of a column may be desirable from the point of view of consistency since in all other cases every occurrence of building component is represented by such an instance in the knowledge base. Due to the large number of such elements - columns, beams etc. - it was not quite feasible. However, taking advantage of the many similarities in these elements, a compromise solution that still retained the refinement of object oriented representation was devised. Using columns as an example, each column object represents columns that have common features. Such common features could be location in a perimeter frame, such as corner location or interior location, resulting in a uniform tributary area for the typical column in the group. In addition to representing a group of columns in plan, the column object incorporates the columns all along the building height by keeping track of all the different sections of the column along the height of the building. The changes in the column sections are made only every 3 or 4 storeys, depending on the total height of the building, keeping in line with the practice of rationalizing structural element sizes for economy of fabrication or in-situ concrete construction.

Similarly for beams, members with the same span and tributary area are represented by a single instance of the beam object. While this method adopted in Tall-D has worked well with the rest of the knowledge representation technique, there are other means in Lisp based systems such as properties and arrays features. While the arrays store sequence of values of the same data type that could be retrieved using numerical indexes, the properties feature permits retrieval based on string indexing. The latter feature could therefore be a memory efficient solution to storing and retrieving large numbers of elements such as columns and beams.

Another issue that involved structural element data was the use of external catalogues such as the CISC standard rolled steel sections. Though the CISC catalogue was not eventually used in-toto in Tall-D system it was used to test the ASCII interface facility of the KBS development tool. The ASCII parser of GoldWorks was set up to read the different types of steel section from the electronic version of the CISC catalogue. Each section was instanced as an object in the knowledge base. This however became untenable due to the large numbers of these sections. Therefore, after experimenting with this frame/instance representation, a subset of the sections data is built-in as part of the program code with only a few properties relevant for preliminary design purposes. However, this exercise proved the ASCII interface for reading any neutral data files. This, along with the geometry export to CAD programs, demonstrates the import/export functionality in Tall-D.

3. 3. 3 Representation of Alternatives

A feature of preliminary design that needed attention is the method of generating and representing multiple solutions or alternatives. The traditional object oriented approach is suited for developing one design from conception to completion. However, generating and maintaining many relevant alternatives simultaneously is not explicitly supported by current data modelling approaches. For example a building configuration may have frame-shearwall structure and braced-frame structure as alternatives. Then, each of those structural alternatives may have different/alternative column spacings or shearwall configurations. Thus there is a need to represent alternatives that are in essence a local variation of one or more aspects of a previous design (although at a lesser level of detail). Some of the features of the KBS development tool were adapted to implement this aspect in Tall-D. Multi-valued slots were used to represent variations of any given design attribute of an object. Additional slots were also used where appropriate to record the presence of an entire class of alternative solutions. Please see section 5.1.1 for a detailed discussion of different structural systems designed by Tall-D and section 5.1.2 for alternative plan layout of such structural systems.

3.4 Formulation of the Knowledge-Base

The objective of the KBS system is to perform multistorey office building design at the preliminary stage taking into account architectural and structural considerations. A study of the design domain at the preliminary stage was required to extract the necessary information for knowledge-base formulation. Interpretation and subsequent formalization was facilitated by the knowledge-base development tool which provided the mechanisms to represent the knowledge after extraction.

Information regarding different elements of the preliminary design process was assembled as:

a) Generation of the overall building configuration;
b) Evaluation criteria for alternative overall building configurations;
c) Approximate cost of the building;
d) Generation of alternative structural schemes for the building;
e) Generation of the structure geometry and loading;
f) Approximate structural analysis and member sizing; and
g) Evaluation criteria for the structural system alternatives.

Information on the above topics was used in the development of the knowledge-base. Various sources were utilised in collecting information for the design system. The knowledge-base presented here does not reflect any particular designer's expertise. Nonetheless, it is an interpretation of the design process as documented in such sources as case histories of building projects, monographs, technical articles in journals and conference presentations, all related to multistorey office building design. Source topics range from overall planning concepts and design of efficient structural systems to suitability for different building configurations and approximate methods of analysis. Of special importance to this system development was the impact of various interactions from different building subsystems on design decisions. Integrated design issues and the ensuing details of implementation were thus given due consideration.

Architectural design concerns are represented in the knowledge-base by including considerations such as the overall building configuration and related evaluation criteria. Subjective information such as the aesthetic value of design alternatives or the effect of adjacent buildings are, however, not considered. The structural engineering domain is simulated by considering the design process of structural systems that are compatible with overall building configurations.

3. 4. 1 Sources of Domain Knowledge

There is a significant amount of information used in the development of the knowledge-base. Most of it is extracted and adapted from a wide range of literature on multistorey buildings, such as case studies analyzed by experienced designers, design guides and textbooks, monograph volumes, journal articles on different projects and state-of-the-art reports. Some of the important sources are given below.

The monograph on tall buildings (Council on Tall Buildings 1978-80) is an authoritative source useful in the development of the knowledge-base. It was produced by committees of experts in different fields of high-rise buildings. Of specific use were the volumes on the topics of planning as well as structural design of steel and concrete multistorey buildings. Discussion of the relative efficiencies of different structural systems is used as part of the basis for the rules in the selection of the steel structural system alternatives. Case studies of already constructed multistorey buildings in both steel and concrete were used to formulate the range of acceptable values for certain parameters in the knowledge-base (Guise 1990). Such parameters include the percentage of usable floor area, percentage of various vertical elements occupying portions of floor area and core. Existing values in various projects thus form the basis of some of the default values (or facts) in the knowledge-base.

Useful sources in the formulation of the knowledge-base for structural design are Taranath (1988) as well as Coull and Stafford-Smith (1991). Illustrations of different structural forms for multistorey buildings and corresponding advantages, disadvantages as well as approximate analysis methods are highlights of these sources. They provide a state-of-the-art in tall building structural systems in use. Response of the lateral structural system is also the subject of a comprehensive ACI report (American Concrete Institute 1982). Behaviour of many of the routinely used structural systems is detailed along with modelling for analysis and member design. Implementation of the preliminary structural design is largely made using these above sources.

Code specifications with respect to the loading (wind, seismic and live loads) and serviceability are available in the National Building Code of Canada (NBCC 1995). Methods of calculating the wind loads and seismic loads are detailed in the supplement to the NBCC (Supplement to NBCC 1995). Minimum design loads for different occupancies are also supplied.

Approximate sizing of members requires consideration of a wide range of structural components and a variety of member forces. Columns, beams and slabs in different materials need to be scaled in a rapid and reasonably accurate fashion for preliminary design. Design charts for the above purpose are taken from Allen and Iano (1990). Other similar sources were consulted, namely, Simplified Design (Portland Cement Association 1992), CISC Steel Handbook (Canadian 1995) and CPCA Handbook for Concrete (Canadian 1995). Cost information is obtained from Yardsticks for Costing (Hanscomb 1995) for the purpose of preliminary estimate.

Some of the heuristics extracted and included in the knowledge-base from the above sources are listed below.

For architectural design:

- The best place for core location to maximise perimeter window access for rental areas is a central location followed by scattered, detached, edge, corner and end locations in that order (Allen and Iano 1989).
- Maximum plan dimension of large multistorey buildings is about 50m to provide reasonable access to exits and services anywhere within the floor, and to ensure that no location is too distant from the window line (Schueller 1986);

- Case studies of rectangular floor configurations of large multistorey buildings show that the maximum dimension of a side is around 85m, with an average maximum of 50m.
- For concrete to be economical there should be enough repetitions. Therefore, with less than ten storeys steel is an economical choice of material (Taranath 1988). The selection of material is however influenced by additional factors such as the soil bearing capacity, availability, duration of construction and designer preference.
- In prismatic office towers of normal floor size (about 3500 m^2), the core is usually planned for one centralized location (Iyengar 1977).
- Buildings that solely house office space have high loads and high storey heights (3.6m to 4.3m or 12ft to 14ft) (Schueller 1986).
- Practical considerations for modular planning of office area require an 8m column-free bay (Holgate 1986).
- The floor area within 7.6m (25ft) to 10.7m (35ft) from the window-line is considered prime rentable space (Khan and Iyengar 1973).

For structural design:

- The bounds on the spans for efficient plane-framed buildings ranges from 6.1m to 15.2m (20ft to 50ft). Since utility of office space decreases with shorter spans, the decrease in rental will have to be verified against savings (Council on Tall Buildings 1979).
- For preliminary design, column sizes are primarily governed by axial loads with appropriate increase in cross-sectional areas for column moments - 10 percent increase for interior columns and 50 percent increase for exterior columns, the later due to the unbalanced vertical load moments (Nilson and Winter 1986).
- In concrete structures, it is more economical to repeat beam sizes and use varying percentages of steel reinforcements than vary the cross-section dimensions. Beam sizes need change only if design moments change 30 to 50 percent or more (Fling 1987).

In addition to the above, other sources have also been used to obtain relevant information about technical details of the design process, as well as the heuristics to develop different groups of rules.

Example of rules, frames and functions used in the system are presented below to illustrate how the information collected is incorporated in the knowledge-base.

3. 4. 2 Examples of Knowledge-base Implementation in Tall-D System

A. Example of an Axiom

An assertion is the building block of a knowledge base. It saves a fact or an axiom in a knowledge base. An example of an assertion would be

```
(Floor_Layout_Generation Done)
```

Such an assertion, for example, can be used by a macro rule to activate the subsequent knowledge module.

B. Example of a Meta-Rule:

```
1  (DEFINE-RULE Actv-Generate-Floor-Rules
2     ( :direction  :forward
3       :sponsor TOP-SPONSOR)
4    (initial-user-interface-rules done)
5    (Use Generate-Floor-Rules)
6
7     THEN
8
9    (evaluate (activate-rule-sets :names
10        '(Generate-Floor-Rules))
11        (forward-chain))
```

Upon initiating the design session the Tall-D system requests information regarding owner requirements for the building project. Upon completion of initial input entry, the knowledge-base considers the assertion shown above in line 4. Whenever the designer selects from the menu bar the *Generate-Floor* option, the second assertion (line 5) is entered in the knowledge-base. When both assertions in the antecedent part of the above rule are entered, the consequent part is executed. The floor generation rules are thus activated for inferencing and forward chaining re-initiated. The overall problem solving strategy thus incorporates such meta-rules to select and activate appropriate groups of rules or rule-sets for specific design tasks.

C. Example of a design Rule

A design rule that is used in the generation of initial outlines is presented below. The rule generates all feasible combinations of length and width values, from dimensions within a predefined range. Ranges for length and width are defined in the object `General-Constraints`. The range is assigned based on the required rentable area.

```
;**Generate Floor Outlines;
(DEFINE-RULE GEN-FP1
   (:print-name "FLOOR-GENERATOR-1"
    :direction  :FORWARD
    :sponsor Generate-floor)
   (INSTANCE  generation-aid is General-Constraints
         with length-range ?l
         with width-range ?w
         with max-floor-length ?ml
         with max-floor-width ?mw)
(>= ?l ?w)
(># ?ml ?l)
(>= ?mw ?w)
(>= 4.0 (/ ?l ?w))
THEN
(evaluate (setq floor-name
          (intern (string (gensym
 'Floor-j))'GW)))
(evaluate  (make-instance   floor-name
        :IS 'Floor-Outline
        :slots `((floor-length  :value  ,?l)
(floor-width  :value  ,?w))))
(evaluate (send-msg (gw-object floor-name)
          :pass-values&draw ?l ?w floor-name)))
```

The antecedent part specifies the name of the rule, direction of chaining, controlling sponsor, binding of variables, and checks to filter out undesirable combinations. Binding of variables is done to select a permissible range of dimensions from an instance `generation-aid` that is an instance of *General-Constraints*. The four checks that follow ensure that:

- length is greater than or equal to width;
- length is less than or equal to maximum permissible value;
- width is less than or equal to maximum value permissible;
- length over width ratio is limited to 4.

The consequent (or THEN) part consists of statements to generate a unique string starting with `FLOOR-`, create an instance of the knowledge representation frame *Floor-Outline* with slot values `floor-length` and `floor-width` set to the values selected in the antecedent part (variables `?l` and `?w` respectively as in lines 7 and 8). The last function in the consequent sends a message to a handler inherited by the floor-outline instance to update the screen and display the instance graphically.

D. Examples of Frames

The following example frames show the implementation details of three individual frames which are part of the knowledge-base. Figure 3.2 shows the hierarchy of frames as arranged in the knowledge-base. During execution of the simulation program instances of the frames as required are generated.

D.1 Frame that acts as the parent frame for all floor layout instances.

```
(DEFINE-FRAME Floor-Outline
   ( :print-name "Floor Outline"
     :doc-string "Frame for instantiation of floor outlines"
    :is Generic-System)
 (floor-length)
 (floor-width)
 (percent-ground-coverage)
 (area-per-floor)
 (floor-area-ratio)
 (Mech-Floors)
 (No-of-floors
   :when-modified (eliminate-Too-Tall))
 (Gross-Area-ofBldg)
 (Bldg-asp-ratio)
 (slenderness-description)
 (core-type :multivalued t)
 (core-instance-names :multivalued t
        :constraints  (:instance-of Core-Config))
 (LatSys-Material-Preference
    :multivalued t)
 (GraSys-Material-Preference
    :multivalued t)
 (Possible-Lateral-Systems
    :multivalued t)
 (Status)
 (Approx-total-Cost)
 (Develop-Structure?)
 )
```

The above frame has many slots that describe the floor outline in increasing detail starting with its length and width, going on to possible structural systems. Constraints are defined as relevant. An example of a demon function is `eliminate-Too-Tall` attached to the `No-of-floors` slot (see line 12 above).

D.2 Frame describing the generic structural system.

```
1  ;; Parent of all planar lateral load resisting frames
2  (DEFINE-FRAME Lateral-Structural-System
3     (     :print-name "Lateral Structural System"
4           :doc-string "Represents a Building Frame"
5       :is Generic-system)
6    (Column-Spacings :multivalued t)
7    (Number-of-Columns)
8    (No-of-Bays)
9    (Typical-Column-Tributary-Area) ;per floor per column
10   (Floor-Outline-Instance)
11   (Structural-Scheme-Column-Layout-Type
12     :constraints (:One-of (Grid-Aligned-to-Core
13                   Perimeter-Based Grid-2D
14                   As-per-shearwalls)))
15   (Frame-Designation)
16   (Type)
17   (Material)
18   (Orientation)
19   (Support-Condition)
20   (Location-in-Plan
21    :constraints (:One-of
22              (In-Front At-Back Right-Edge
23                Left-Edge Perimeter Inside)))
24   (Column-Instances :multivalued t
25          :constraints
26          (:instance-of Column-Element))
27   (Displayed
28     :default-values (NO))
29   (Orthogonal-Counterpart
30     :constraints (:Instance-of Lateral-Structural-System))
31   (Column-Tributary-Area-Interior-Row-XX)
32   (Column-Tributary-Area-Interior-Row-YY)
33   )
```

This object inherits from *Generic-System*, which is defined as a parent (line 5). It in turn is the parent of many individual structural system objects such as rigid frame, braced frame, frame-shearwall interaction etc. Instances of these frames are the stability element in a structural scheme like the one described by the object in next listing shown in D.3 (*Frame-ShearWall-Scheme*). See lines 4, 5 and 6 of the listing in D.3 below. The result of inheritance of attributes (or slots) across three objects makes the object at the bottom of the hierarchy specialised to the required extent. Attributes can also be modified at the lower end of the hierarchy and have the local slot constraints merge with that inherited if needed to fit the local context. A few of the slots are repeated in this frame for clarity: eg. `Orientation` was defined in *Generic-System* with appropriate constraints.

D.3 Frame with slots describing a frame shearwall structural system.

```
(Define-Frame Frame-ShearWall-Scheme
  ( :Print-Name "Frame ShearWall Scheme"
 :is Super-Structure)
 (Stability-Element
     :Constraints
     (:One-of (Frame-ShearWall)))
 (SWall-Direction        ;;Primary direction of S/W
   :constraints (:one-of (One Two)))
 (General-Location-of-Walls
   :multivalued t
   :constraints (:one-of (In-Core Front
                  Back Left Right Centre)))
 (Approximate-Number-of-Walls-XX) ;Wall axis prll to XX
 (Approximate-Number-of-Walls-YY) ;Wall axis prll to YY
 (Uniform-Spacing-XX-Walls) ; in metres
 (Uniform-Spacing-YY-Walls) ; of walls w/axis pll. to YY
 (ShearWalls
   :multivalued t
   :constraints (:instance-of ShearWall-Element))
 (Uniform-Length-of-XX-Walls)
 (Uniform-Length-of-YY-Walls)
 (Composite-Construction
     :Constraints
     (:One-of (Possible Not-Possible)))
             ; IF Possible, means steel columns
             ; used with concrete shearwalls
             ; as an alternative to concrete cols.
 (Steel-Used-In :multivalued t
     :Constraints
     (:One-of (External-Columns Internal-Columns)))
 (Column-Material :Default-values (Concrete ))
 (Typical-SW-Frame-XX
   :constraints (:Instance-of Frame-Shearwall))
  )
```

This frame describes the shearwall structural scheme. Combined with the inherited attributes and the attributes of related objects, objects similar to this one but corresponding to the respective structural systems enable the development of a comprehensive description of structural components by the dynamically generated instances. The rules in the respective rule-sets of the knowledge-base instantiate and populate these objects with valid attribute values.

E. Examples of Functions

E.1 Demon Function

A demon function attached to a slot of frame triggers upon an event related to that slot. The event could be such thing as a modification of the slot-value. The following list shows one such function in the `No-of-floors` slot of *Floor-Outline* (see listing in C.1 in previous section). This function eliminates building alternatives with storeys more than owner specified as well as with storeys less than 5, by setting the value of the `status` slot to `Del-N-F` (see line 7 below). A message to the effect is issued to the user (lines 10 to 18). If the number of storeys is in the valid range then the status is set to `OK-N-F` (line 21). These functions have access to the instance, slot, as well as to the old and new slot values (see line 1).

```
1  (defun eliminate-too-tall (inst slot old-val new-val)
2    (if (OR
3        (< (slot-value 'owner-constraints 'max-no-of-floors)
4                     (car new-val))
5        (>= 5 (car new-val)))
6        (Progn
7           (Setf (slot-value inst 'status) 'Del-N-F)
8           (format t "~&~A will be deleted; No. of floors: "
9                              (Gw-Name Inst))
10          (Let* (
11             (Output (Gw-object 'information))
12             (MyList
13                (format nil
14                 "~&~A will be deleted; No. of floors: ~D"
15             (Gw-Name Inst) (car new-val))))
16             (Set-slot-value Output 'Title
17                "Warning: Floor Configuration")
18             (Set-slot-value Output 'Display Mylist)
19             )
20          ))
21    (Setf (slot-value inst 'status) 'OK-N-F)
22    )
23 )
```

E.2 Example of Handler Function

The handler function is a member of an object. It can only be invoked by sending a message to an instance of the object of which the function is a member. When invoked the function has access to the instance through the variable `self`. The example function below generates building lateral loads according to NBCC (1995). Since a uniform lateral load of intensity equivalent to the value at top-third of the building produces approximately similar forces in the structure as that produced by the near triangular variation obtained by a more rigorous application of the equations, the former is preferred due to the resulting simplification.

```
;; External pressure/suction P = qCeCqCp
;; q =  [=0.37 KPa for Dorval 1/30 prob]
;; Ce exposure factor, value from
;;    table 4.1.8.A (0.9 to 2.0) based on H
;; Cg gust factor
;; Cp external pressure coeff. averaged over the area
(define-handler (Building-Configuration
             Generate-Lateral-Loads) ()
  (Let*(
     (Height (slot-value Self 'Building-Height))
     (q (slot-value 'Building-Loads-Code
                'Hourly-wind-pressure))
     (H23 (/ Height 1.5)) ; 2 thirds height of building
     ;Ce at top third height of building
     (Ce (expt (floor (/ H23 10.0)) 0.2))
     ;actually varies along the height of building.
     ; To get continuous variation use the above
     ; expression, substitute H23 by the corresp.
     ; elevation(m)
     (Ce (Cond
      ((< Ce 0.9) 0.9)
      (t Ce)) )
     (Cg 2.0) ;for buildings as a whole; 2.5 for clad.
     (CpWin 0.8) ; Cp on wind-ward side
     (CpLee 0.5) ; Cp on lee-ward side
     ;;(P_Win_H1 (* q Ce Cg CpWin)) Not used
     (P_Win_Abv_H1 (* q Ce Cg CpWin)) ;Current: Unfm.dist.
        ; top-third value assume uniform due to
        ; approx. the same effect caused at the base.
        ; Use Ce from table 4.1.8A as per
        ; Z-elevation of point on the building,
        ; if continuous var. required.
     (P_Lee (* q Ce Cg CpLee))
     )
    (Print (list "P_Win_Abv_H1 P_Lee (in kPa):"
                 P_Win_Abv_H1 P_Lee))
    (set-slot-value Self 'Windward-Pressure-at-Top3dHeight
                 P_Win_Abv_H1) ;kPa
    (set-slot-value Self 'Leeward-Pressure-at-Top3dHeight
                 P_Lee)  ;kPa
    )
  )
```

D.3 Example of Direct Function

There is another type of function that is general purpose. Once defined it can be called from any other part of the knowledge-base: consequent part of a rule, another function, etc. The following function calculates the aspect ratio for the building and prompts the deletion of the alternative if that value exceeds 10, by setting the `status` value to `Del-BSR` (line 16). The aspect ratio calculated is placed as the slot value in `bldg-asp-ratio` (lines 8 to 11).

```
1  ; For rule Aspect-Ratio-bldg
2  (defun   cal-bldg-asp-ratio
3      (Ins No-of-fl fl-length fl-width fl-height Stat)
4    (let*  ((HB (* no-of-fl fl-height))
5    (hbl (/ HB fl-length))
6    (HBW (/ HB fl-width)))
7    (progn
8     (If (eql stat 'OK-N-F)
9        (if (> HBL HBW)
10          (set-slot-value Ins 'bldg-asp-ratio HBL)
11       (set-slot-value Ins 'bldg-asp-ratio HBW)
12       )
13    )
14    (If (and (or (> hbl 10) (> hbw 10))
15        (eql (slot-value Ins 'status) 'Ok-n-f))
16       (setf (slot-value Ins 'status) 'Del-BSR))
17   )))
```

3. 5 External Interfaces in Tall-D

Integration of architectural and structural domains for preliminary design being the main scope, interfacing with external programs was not a requirement in achieving the project objectives. However as discussed in section 2.3 and 2.4, one of the demands on design systems is the ability to exchange design data, not the least of which is geometric data. Moreover, the results of preliminary design should, in the normal process, lead to more detailed design.

One of the methods of exchanging data is to use predefined file formats. Such standards for the interchange of information have been discussed in section 2. 4. 3. Tall-D provides two interfaces to demonstrate the need and utility of such interfaces - one for reading in the section properties of numerous steel sections and another interface for the export of geometry of specific Tall-D design alternatives.

3. 5. 1 Interface for 3D Geometry Data Exchange

Tall-D provides a data interface to many CAD packages that support the DXF format of AutoCad. Using the DXF interface of Tall-D, it is possible to use commonly available CAD packages to view and work with specific Tall-D design alternatives. The simple interface generates a text file that contains a three dimensional wire-frame data of the building structure. An example of such a wire-frame view obtained using this interface is shown in Fig. 3.2. The two wireframes in the figure were plotted after importing the same file in two different CAD packages - *Microstation* and *AutoCad.* The diagram on the left the figure is produced with the former. The one on the right is produced with the latter after removing hidden lines. Thus the figure is an illustration that the interface is functional with two leading CAD packages and, in fact, all packages supporting the DXF format. An extension of this feature could implement the evolving STEP standard to not only provide the geometry as in the current system, but include design information as well for programs able to import and use such data.

3. 5. 2 Interface for CISC Steel Data Files

The KBES development tool GoldWorkII provides an ASCII parser that can be used to read data files if the format of the file is known. The Canadian Institute of Steel Construction (CISC) section properties for hundreds of different rolled steel sections is available in ASCII files. This data is parsed and input to Tall-D system to set up the tables for the steel structural design. The use of these sections in the design was deferred after implementing a version where each section was instantiated as a *Structural-Element.* Due to the number of sections represented as an object, the system slowed and stalled. Though alternative methods of accessing the CISC data, such as using a database resident on the secondary storage or even a memory resident array of numbers were available, it became clear the use of individual steel sections was more in the scope of detailed design. Due to the chart-based method of preliminary design for steel structural elements sizing used in Tall-D, only a sub-set of steel sections and their few key data were needed, and hence included within the Lisp function performing such steel sizing.

3. 6 Summary

An overview of the system architecture of Tall-D has been presented in various detail. The size of the knowledge base was described by the system metrics. The object oriented knowledge representation of the physical building was illustrated in detail. This along with the rule base constitutes the hybrid knowledge representation scheme used in Tall-D. The sources of information used to formulate the knowledge-base have been mentioned. Example program code showing the different types of knowledge-representations as well as procedural routines was presented. An external interface in Tall-D, that enables export of building geometry, was illustrated.

In the next chapter, the system is described in the context of the development of the overall building configuration, the step prior to structural systems design.

CHAPTER 4

Development of Overall Building Configuration

A detailed description of the components of the KBS related to architectural design is presented. This includes details regarding the overall building configuration as well as the initial cost estimation procedure by means of a demonstrative example.

4. 1 Overall Building Configuration

The overall layout planning of a building at an early stage of design ensures that requisite space for the different building components is allocated, satisfying at the same time functionality and flexibility. Competing objectives and needs such as optimal location of the structural system, maximisation of prime rentable area, provision of appropriate space and connections for various mechanical system components and other building services have to be addressed and fulfilled.

Overall building configuration specifically deals with building massing, where the location of the different component areas of the building both in plan and elevation are the major decisions. The plan layout significantly influences the final efficiency or acceptability of the design. It also greatly influences the vertical structure with respect to its location and the kind of system. The total number of storeys and floor area determine the location of the mechanical plant. The system however does not consider structure configuration below the ground level nor the presence of a plaza at the street level.

The overall building configuration module described here constitutes the architectural module of the integrated building design KBS for the design of multistorey office buildings at the preliminary stage. In the following, a discussion of layout planning for multistorey buildings is presented, followed by the adopted methodology for the overall configuration of buildings. An example of a typical layout configuration performed with the assistance of the KBS is presented and progressively developed in the different subsections. Consideration of the structural system alternatives is presented in the subsequent chapter.

4. 2 Floor Layout

Building floor layout problem solving involves selecting and arranging component areas to form a complete floor. This task requires knowledge about various space requirements and inter-relations between the component areas. In typical building layout problems, there are numerous feasible solutions, thus indicating the need for a robust selection process to search through alternative solutions. Numerical computations then appear to be of secondary importance since the determination of an appropriate space and location for each component is not a process suited to a pure algorithmic description.

Determination of the spatial needs of different building subsystems (like structure, mechanical, interior etc.) requires consideration of the services in the building and functions of the subsystems. To achieve efficient solutions to space allocation problems, it is necessary to take advantage of the multiple needs the same space can cater to, e.g. a core in which all the services and shafts are grouped together can be used to place the lateral load-resisting structural system as well. Such cores built of concrete also serve as fireproof enclosures to house staircases as well as fire-resistant separations between areas on the floor, dividing occupant areas into compartments with respect to fire-safety regulations (NBCC 1995). In a similar fashion a suspended ceiling can house ducts, structural beams and lighting fixtures. The simultaneous consideration of diverse needs for different components and subsystems in buildings (Rush 1986) is effective in achieving building integration, and is most relevant at the early design stage.

Layout planning begins with the generation of simple outline alternatives that are progressively refined by defining component areas on the plan. The initial outlines are determined by the designer conforming to site constraints, Code regulations and owner requirements. As details are developed, the number of initial outline alternatives decreases rapidly since those alternatives violating design constraints are eliminated.

Conventional computer-based systems to automate the above process begin with an algorithmic generation of alternatives, followed by a comprehensive search for the best solution. They require as input the building geometry and extensive definition of constraints to go through the process of generation and evaluation of solutions. Such systems are generally restricted to one type of geometric figure and to a narrow range of permissible dimensions, thus limiting the number of alternatives that can be examined and the quality of the final solution.

In the Tall-D system presented here to develop overall configurations, a more flexible approach is adopted following the hierarchical generate-test technique implemented in a KBS environment (Hayes-Roth et. al. 1983). Improvements to this method compared to the conventional algorithmic method are based on developing a design context to define better generation constraints, to generate solutions incrementally and to evaluate them at each hierarchical level in order to retain only the most promising alternative at that level. Such procedure with multiple levels of generate-test is effective in keeping the number of alternatives in check. Fig. 4.1 summarizes the complete development of a building structural layout alternative from information available to the building designer at the preliminary stage.

The first level of description consists of the rectangular shape of floor, the number of storeys and the corresponding building slenderness ratio. Each alternative at this level gives rise to a group of alternatives at the subsequent level with different core configurations (Fig. 4.1). Similarly at the second level of description, each alternative leads to a group of alternatives at the subsequent level with different structural configurations. The concept of multiple levels of abstraction, each focusing on a different

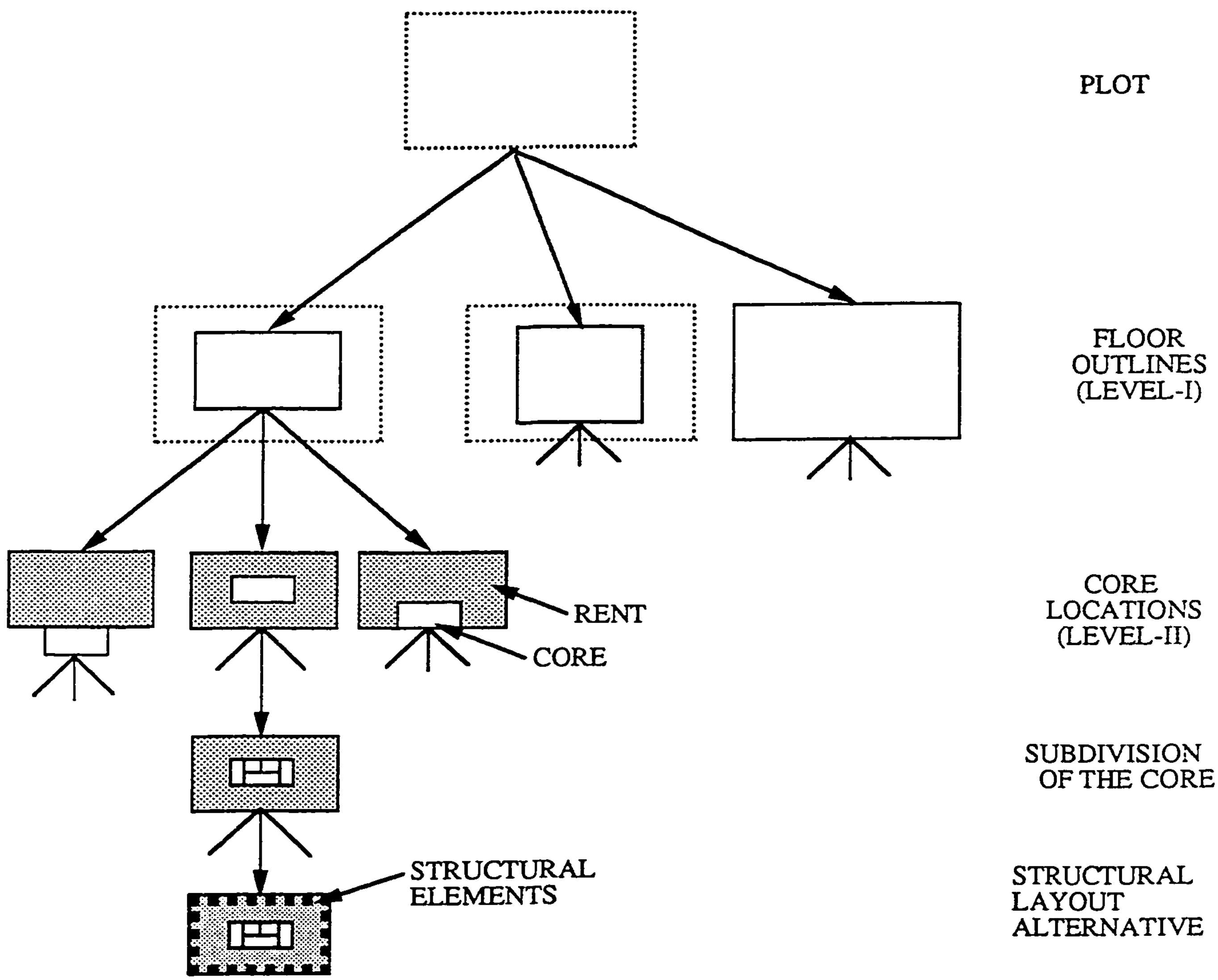


Figure 4.1 Development of a Building Structural Layout at Different Levels of Details

level of detail has proven to be effective in reducing the complexity of large problems (Korf 1987). A similar concept of hierarchical modelling has previously been proposed for architectural design layout problems (Willey 1978).

However this procedure requires that domain knowledge in the form of heuristics and constraints be integrated with the generate-test cycles. The KBS approach facilitates the incorporation of constraints from Codes and zoning laws as well as design heuristics derived from the technical literature and case studies. Practical considerations like rentability, placement of structural systems for lateral load resistance as well as allocation of space for service area, circulation, elevators and mechanical risers can then be accounted for simultaneously within generate-test cycles.

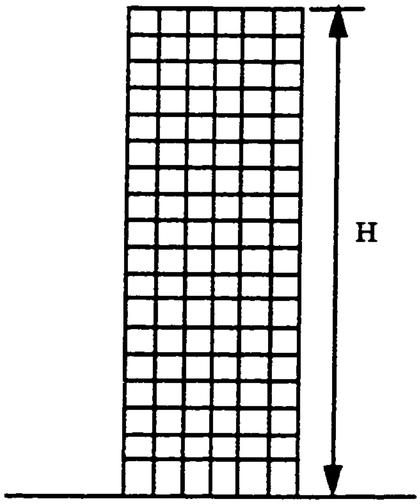
The KBS presented here is restricted to plots and building floor outlines of rectangular shape, with office occupancy applicable to the entire rentable area. The number of storeys is limited to fifty. In high-rise buildings up to twenty storeys, constructional features are generally repetitive with respect to height (Barton 1983). However in taller buildings, the number of elevators reduce towards the top of the building (White and Salmon 1987), thus leading to additional usable space in the top floors. Since all the vertical services including the elevators are grouped together, the variations in elevator banks do not affect significantly the functionality of the designs based on a typical floor at the lower level. Hence for simplicity and preliminary design purposes, the configuration of only one typical floor is considered here.

4. 3 Generation of Configuration Alternatives

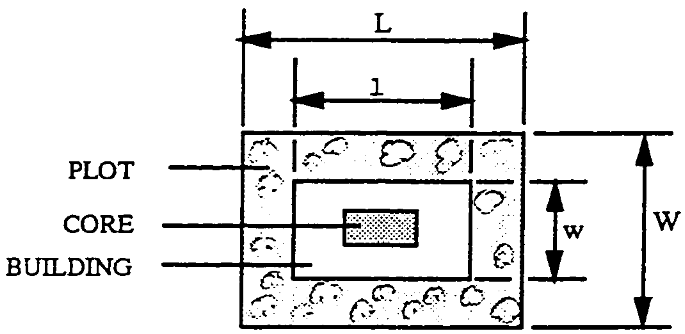
Figure 4.2 illustrates the basic definitions of terms used in the following discussion in terms of building geometric parameters and ratios. The floor area ratio (FAR) relates total floor area in the building to the plot area. Permissible FAR is associated with percentage of ground coverage and is specified by municipalities as part of the zoning by-laws. Higher FAR limits usually are permitted if a lower percentage of the plot is covered by the building footprint. The building slenderness ratio (BSR) is routinely used in structural design against lateral loads. The floor aspect ratio (FAS) characterises the shape of a typical floor outline.

Figure 4.3 shows a schematic of the adopted generate-test procedure. The tree or hierarchy of design alternatives illustrates the concept of generate-test at multiple levels as described above. Comments on the left of the tree indicate the sequence and type of operations performed, and text on the right indicates the corresponding modules of information either used or generated by the system. The tree itself represents the design space as the procedure advances from top to bottom i.e. the generation and gradual reduction of initial outline alternatives through Level-I, followed by a new generate-test cycle through Level-II.

The design context is initially set up by the owner specifications specified in Figure 4.3. An example of owner specifications is given in Part-A of Table 4.1. The process begins with the generation of initial outline alternatives with details concerning rentable space, number of floors and slenderness ratio of the building. The first generation constraints to be used are related to the practical range of floor dimensions and the permissible FAR for different percentages of ground coverage. The range of dimensions to be considered in the generation of outlines is set using the required floor area as an indicator. The generation of initial floor outlines of rectangular shape is carried out within these constraints and limited to a floor aspect ratio (FAS) of four (4) as most buildings fall within that limit. From the energy consumption point of view, this limit also ensures that the external surface of the building envelope remains within practical values with



ELEVATION



PLAN

Number of Storeys	= N	
Floor Area Ratio (FAR)	= Total Floor Area / Total Plot Area	
	= (N * w * l) / (W * L)	
Floor Aspect Ratio (FAS)	= l/w	
Building Slenderness Ratio (BSR)	= H/w	valid if $L \geq W$
		and $l \geq w$

Figure 4.2 Basic Geometric Parameters and Ratios.

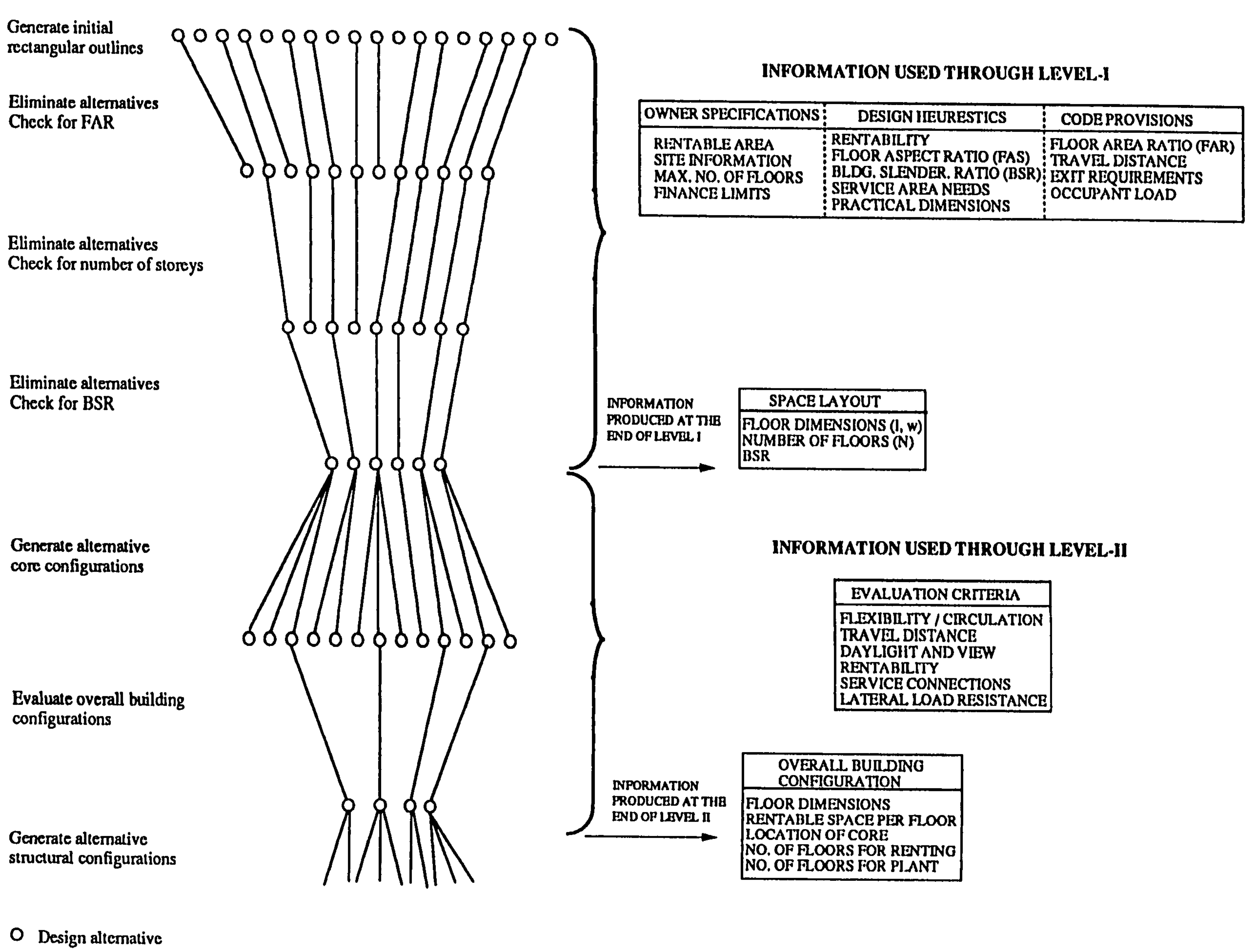


Figure 4.3 Generate-Test Procedure as used in the KBS to Configure Multistorey Buildings

respect to the enclosed space to heat or cool. These alternatives are generated breadth-first fashion i.e. with a single detail being developed for all alternatives, then a second detail and so on. The generation of each detail is followed by a test against a specific constraint. For example after the generation of the floor dimensions, the percentage of ground coverage is calculated and used to check against the permissible FAR. The outlines that require to exceed the FAR limit in order to provide the required rental area are eliminated. Elimination of alternatives is also performed after the number of storeys is calculated as well as after the BSR is calculated. The above three eliminations complete Level I of the generate-test procedure. Constraints based on design heuristics and Code regulations that are used to narrow down the design space to a relevant set of overall configurations are shown in Fig. 4.3. Level II of the generate-test procedure is concerned with the generation of alternative core locations for the current outlines retained at the end of Level I (see also Fig. 4.1). The location of the service core in the floor layout is an important design decision since it affects rentability in addition to a number of other building characteristics. As previously mentioned, the core is often used to locate the lateral load-resisting system. However the location of the core is also influenced by the size and shape of the floor outline. The provision of a central core leaves maximum space available for rent near the window line. If this is not possible, alternative core placements shown in Fig. 4.4. like edge, detached, two central, corners or end types are considered so as to maximise contiguous usable space on the floor.

4. 3. 1 Evaluation of Overall Configurations

The distribution of domain knowledge between generator and evaluator represents an important feature of a design KBS based on the generate-test paradigm. Heuristics are incorporated in the generator to make the generation process efficient and to limit the evaluation process only to a reduced set of potential solutions. The evaluation part of the system incorporates domain heuristics to test the performance of design alternatives against a set of criteria. The number of generic layout alternatives is reduced going through successive cycles of generate-test, at the same time producing new variations of those that are retained as finer details are developed. Through Level I, generation of

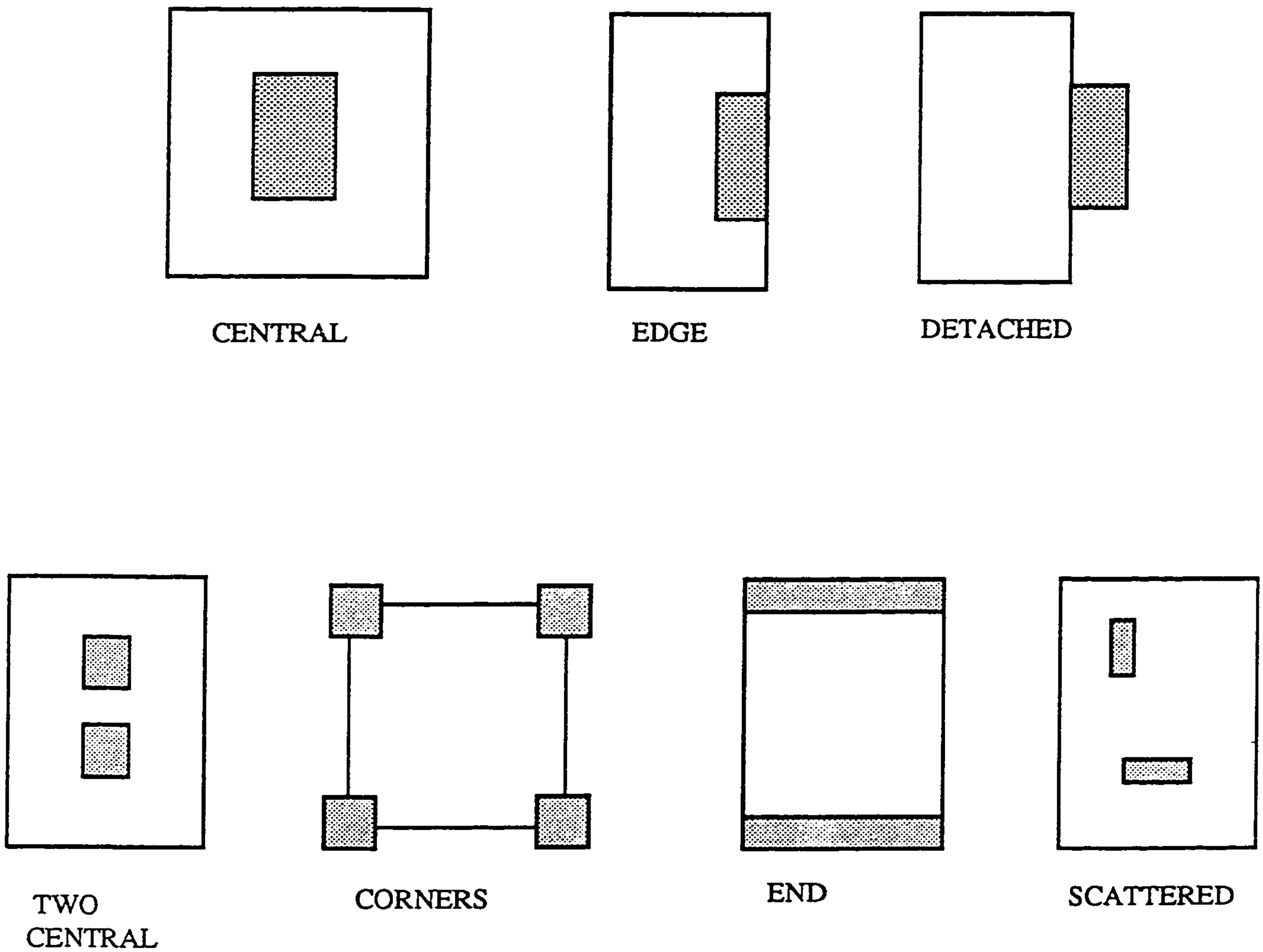


Figure 4. 4. Generic Types of Core Placement Considered in the KBS.

alternatives each with specific building parameters like total floor area, number of storeys, building slenderness ratio etc. is followed by a test against a corresponding constraint. Evaluation of the overall building configurations is carried out in Level II after core locations are added to the floor outlines. The criteria considered (Allen and Iano 1989) in the evaluation of these overall configurations are listed below:

- Flexibility of rental areas;
- Window line access for rental areas;
- Suitability for lateral structural system;
- High rentability at ground level;
- Travel distance from core to window-line;
- Clarity of circulation within rental areas;
- Daylight and view for core areas;
- Service connections at roof level;
- Service connections at ground level;
- General energy efficiency potential.

The type of core placement has significant influence on the above evaluation. For example central core location (either one core or two central cores) is preferable to other types given in Fig. 4.4 for locating the lateral load resisting structural system. To establish a distinction between the generic types of core placement shown in Fig. 4.4, relative ratings are assigned with regard to each of the above evaluation criteria on a scale of 1 to 5, with 5 being the best rating and 1 the worst. To complete the evaluation of overall configurations at the end of Level II, preferences are requested from the designer (see Figure 4.5) against each evaluation criterion and used to weigh the relative ratings. The sum of weighted ratings thus obtained is used to rank the overall configurations against each other, the most desirable being the one with the maximum aggregated value. Based on this evaluation, a ranked list of feasible layouts is finally presented to the designer to enable him to proceed with one (or few) of these to the generation of structural system alternatives.

4. 3. 2 A Typical Design Session

A typical design session starts with input of general design requirements provided by the owner. These are the plot dimensions, maximum number of floors and required floor area as shown in Table 4.1, Part-A. Initial floor outline alternatives are first generated by the system (Part-B) and checked against allowable FAR. Floor dimensions, permissible floor area ratio, number of floors and slenderness ratio of the building are part of the information generated by the system. The alternatives retained in the system at the end of Level-I are given in Part-C, a total of 27 from the 42 initially generated. All the initial alternatives satisfy the FAR regulation, 12 exceed the maximum number of storeys specified and 3 exceed the BSR limit.

For each outline retained at the end of Level-I, an attempt is made to generate compatible core configurations, thus leading to a total of 32 overall building configurations in Part-D. Figure 4.6 shows these overall configurations schematically, as displayed by the system.

Design preferences for the evaluation of overall configurations are specified by the designer as shown in Part-E and in Figure 4.5 on a 0 to 100 scale (0: not relevant, 100: overriding importance). Following this evaluation, overall configurations are priority-ranked relative to each other and presented to the designer (Part-F). Three alternatives, Floor-1, Floor-14 and Floor-25, obtain top score and ranked number one. It may be noted that all three have a central core and are compatible with the preferences given by the designer.

In the end, the designer may select any of the feasible alternatives listed in Part-F, or let the system select the first ranked alternatives, in order to proceed with the development of compatible structural system alternatives, which is described in the next chapter.

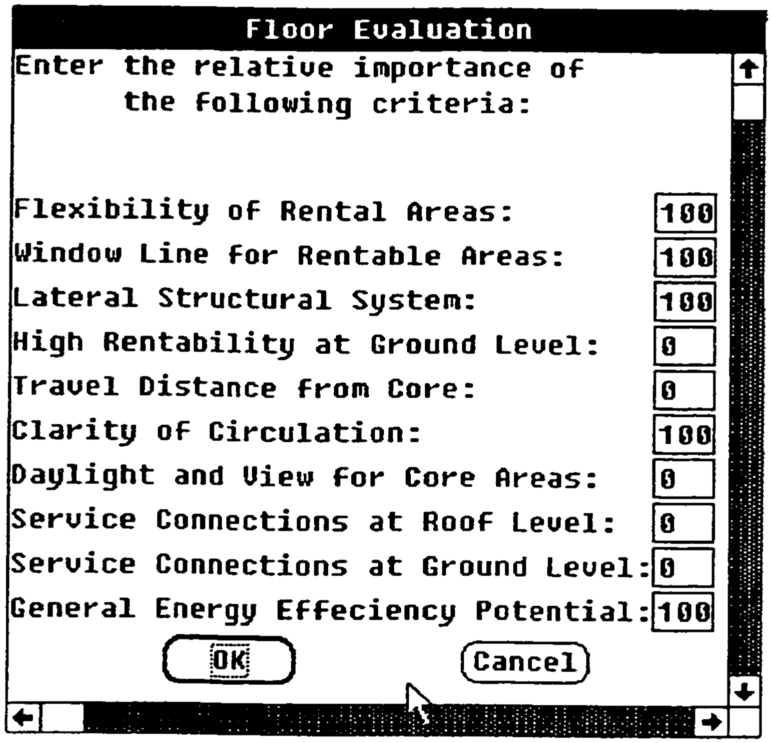


Figure 4.5 Floor evaluation preferences supplied by the designer

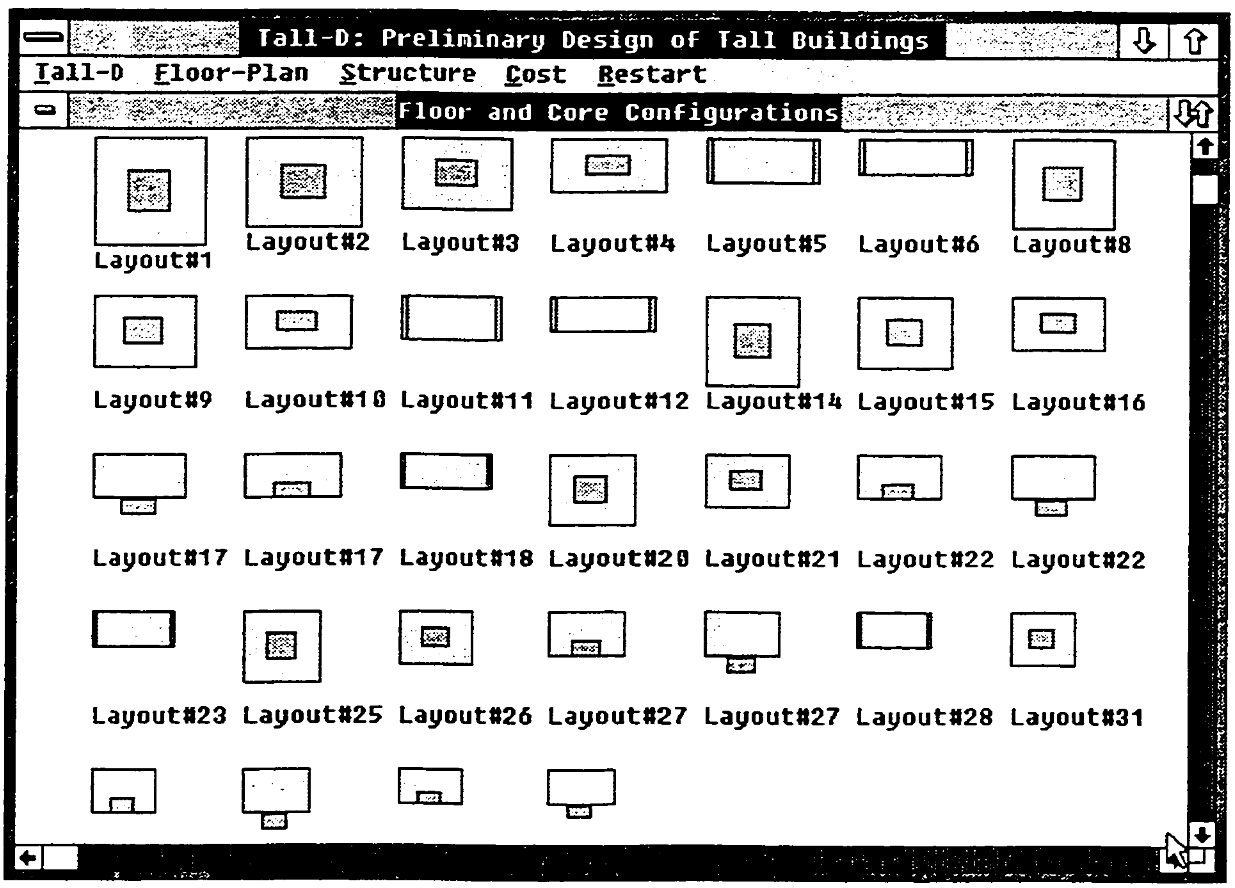


Figure 4.6 Plan view of alternative floor outlines.

Table 4.1 Typical Design Session- Overall Building Configuration.

Part-A: General Design Requirements	
Name of the Building	Example
Length of Plot	80 m
Width of Plot	70 m
Finance Limit	50 million
Max. no. of Floors	50
Required Floor Area	28000 sq. m

Table 4.1 Typical Design Session- Overall Building Configuration(continued).

Part-B: Initial Floor Outlines Considered				
Outline Alternative	Length (m)	Width (m)	Permissible FAR	Status
FLOOR-1	60	60	12	OK-FAR
FLOOR-2	60	50	15	OK-FAR
FLOOR-3	60	40	15	OK-FAR
FLOOR-4	60	30	18	OK-FAR
FLOOR-5	60	25	18	OK-FAR
FLOOR-6	60	20	18	OK-FAR
FLOOR-7	60	15	18	OK-FAR
FLOOR-8	55	50	15	OK-FAR
FLOOR-9	55	40	18	OK-FAR
FLOOR-10	55	30	18	OK-FAR
FLOOR-11	55	25	18	OK-FAR
FLOOR-12	55	20	18	OK-FAR
FLOOR-13	55	15	18	OK-FAR
FLOOR-14	50	50	15	OK-FAR
FLOOR-15	50	40	18	OK-FAR
FLOOR-16	50	30	18	OK-FAR
FLOOR-17	50	25	18	OK-FAR
FLOOR-18	50	20	18	OK-FAR
FLOOR-19	50	15	18	OK-FAR
FLOOR-20	45	40	18	OK-FAR
FLOOR-21	45	30	18	OK-FAR
FLOOR-22	45	25	18	OK-FAR
FLOOR-23	45	20	18	OK-FAR
FLOOR-24	45	15	18	OK-FAR
FLOOR-25	40	40	18	OK-FAR
FLOOR-26	40	30	18	OK-FAR
FLOOR-27	40	25	18	OK-FAR
FLOOR-28	40	20	18	OK-FAR
FLOOR-29	40	15	18	OK-FAR
FLOOR-30	40	10	18	OK-FAR
FLOOR-31	35	30	18	OK-FAR
FLOOR-32	35	25	18	OK-FAR
FLOOR-33	35	20	18	OK-FAR
FLOOR-34	35	15	18	OK-FAR
FLOOR-35	35	10	18	OK-FAR
FLOOR-36	25	25	18	OK-FAR
FLOOR-37	25	20	18	OK-FAR
FLOOR-38	25	15	18	OK-FAR
FLOOR-39	25	10	18	OK-FAR
FLOOR-40	20	20	18	OK-FAR
FLOOR-41	20	15	18	OK-FAR
FLOOR-42	20	10	18	OK-FAR

Table 4.1 Typical Design Session- Overall Building Configuration(continued).

Part-C: Status at the end of Level-I					
Outline Alternative	Height (m)	BSR	No. of Storeys	Mechanical Floors	Status
FLOOR-1	40.0	0.7	10	1	OK-N-F
FLOOR-2	48.0	1.0	12	1	OK-N-F
FLOOR-3	60.0	1.5	15	1	OK-N-F
FLOOR-4	76.0	2.5	19	1	OK-N-F
FLOOR-5	92.0	3.7	23	1	OK-N-F
FLOOR-6	112.0	5.6	28	1	OK-N-F
FLOOR-7	NIL	10.4	39	2	DEL-BSR
FLOOR-8	52.0	1.0	13	1	OK-N-F
FLOOR-9	64.0	1.6	16	1	OK-N-F
FLOOR-10	84.0	2.8	21	1	OK-N-F
FLOOR-11	100.0	4.0	25	1	OK-N-F
FLOOR-12	124.0	6.2	31	1	OK-N-F
FLOOR-13	NIL	11.2	42	2	DEL-BSR
FLOOR-14	56.0	1.1	14	1	OK-N-F
FLOOR-15	68.0	1.7	17	1	OK-N-F
FLOOR-16	92.0	3.1	23	1	OK-N-F
FLOOR-17	108.0	4.3	27	1	OK-N-F
FLOOR-18	140.0	7.0	35	2	OK-N-F
FLOOR-19	NIL	12.3	46	2	DEL-BSR
FLOOR-20	76.0	1.9	19	1	OK-N-F
FLOOR-21	100.0	3.3	25	1	OK-N-F
FLOOR-22	120.0	4.8	30	1	OK-N-F
FLOOR-23	156.0	7.8	39	2	OK-N-F
FLOOR-24	NIL	NIL	51	NIL	DEL-N-F
FLOOR-25	88.0	2.2	22	1	OK-N-F
FLOOR-26	112.0	3.7	28	1	OK-N-F
FLOOR-27	140.0	5.6	35	2	OK-N-F
FLOOR-28	172.0	8.6	43	2	OK-N-F
FLOOR-29	NIL	NIL	57	NIL	DEL-N-F
FLOOR-30	NIL	NIL	85	NIL	DEL-N-F
FLOOR-31	132.0	4.4	33	2	OK-N-F
FLOOR-32	160.0	6.4	40	2	OK-N-F
FLOOR-33	196.0	9.8	49	2	OK-N-F
FLOOR-34	NIL	NIL	66	NIL	DEL-N-F
FLOOR-35	NIL	NIL	97	NIL	DEL-N-F
FLOOR-36	NIL	NIL	55	NIL	DEL-N-F
FLOOR-37	NIL	NIL	69	NIL	DEL-N-F
FLOOR-38	NIL	NIL	91	NIL	DEL-N-F
FLOOR-39	NIL	NIL	135	NIL	DEL-N-F
FLOOR-40	NIL	NIL	85	NIL	DEL-N-F
FLOOR-41	NIL	NIL	113	NIL	DEL-N-F
FLOOR-42	NIL	NIL	168	NIL	DEL-N-F

Table 4.1 Typical Design Session- Overall Building Configuration (continued).

Part-D: Core Configuration

Outline Alternative	Length	Width	FAR	BSR	Storeys	Building Slenderness	Status	Core Type	No. of Cores	Core Length	Core Width	X co-ordinate of core corner	Y co-ordinate of core corner
FLOOR-1	60	60	12	0.7	10	MHS	OK-N-F	CENTRAL	1	23.2	23.2	18.4	18.4
FLOOR-2	60	50	15	1.0	12	MHS	OK-N-F	CENTRAL	1	23.2	19.4	18.4	15.3
FLOOR-3	60	40	15	1.5	15	MHS	OK-N-F	CENTRAL	1	23.2	15.5	18.4	12.3
FLOOR-4	60	30	18	2.5	19	MHS	OK-N-F	CENTRAL	1	23.2	11.6	18.4	9.2
FLOOR-5	60	25	18	3.7	23	MHS	OK-N-F	ENDS	2	4.5	25.0	0.0	0.0
FLOOR-6	60	20	18	5.6	28	SLN	OK-N-F	ENDS	2	4.5	20.0	0.0	0.0
FLOOR-8	55	50	15	1.0	13	MHS	OK-N-F	CENTRAL	1	21.3	19.4	16.8	15.3
FLOOR-9	55	40	18	1.6	16	MHS	OK-N-F	CENTRAL	1	21.3	15.5	16.8	12.3
FLOOR-10	55	30	18	2.8	21	MHS	OK-N-F	CENTRAL	1	21.3	11.6	16.8	9.2
FLOOR-11	55	25	18	4.0	25	MHS	OK-N-F	ENDS	2	4.1	25.0	0.0	0.0
FLOOR-12	55	20	18	6.2	31	VSLN	OK-N-F	ENDS	2	4.1	20.0	0.0	0.0
FLOOR-14	50	50	15	1.1	14	MHS	OK-N-F	CENTRAL	1	19.4	19.4	15.3	15.3
FLOOR-15	50	40	18	1.7	17	MHS	OK-N-F	CENTRAL	1	19.4	15.5	15.3	12.3
FLOOR-16	50	30	18	3.1	23	MHS	OK-N-F	CENTRAL	1	19.4	11.6	15.3	9.2
FLOOR-17	50	25	18	4.3	27	MHS	OK-N-F	EDGE	1	19.4	9.7	15.3	0.0
								DETACHED	1	19.4	9.7	15.3	-9.7
FLOOR-18	50	20	18	7.0	35	VSLN	OK-N-F	ENDS	2	3.7	20.0	0.0	0.0
FLOOR-20	45	40	18	1.9	19	MHS	OK-N-F	CENTRAL	1	17.4	15.5	13.8	12.3
FLOOR-21	45	30	18	3.3	25	MHS	OK-N-F	CENTRAL	1	17.4	11.6	13.8	9.2
FLOOR-22	45	25	18	4.8	30	SLN	OK-N-F	DETACHED	1	17.4	9.7	13.8	-9.7
								EDGE	1	17.4	9.7	13.8	0.0
FLOOR-23	45	20	18	7.8	39	VSLN	OK-N-F	ENDS	2	3.4	20.0	0.0	0.0
FLOOR-25	40	40	18	2.2	22	MHS	OK-N-F	CENTRAL	1	15.5	15.5	12.3	12.3
FLOOR-26	40	30	18	3.7	28	MHS	OK-N-F	CENTRAL	1	15.5	11.6	12.3	9.2
FLOOR-27	40	25	18	5.6	35	SLN	OK-N-F	DETACHED	1	15.5	9.7	12.3	-9.7
								EDGE	1	15.5	9.7	12.3	0.0
FLOOR-28	40	20	18	8.6	43	VSLN	OK-N-F	ENDS	2	3.0	20.0	0.0	0.0
FLOOR-31	35	30	18	4.4	33	MHS	OK-N-F	CENTRAL	1	13.6	11.6	10.7	9.2
FLOOR-32	35	25	18	6.4	40	VSLN	OK-N-F	DETACHED	1	13.6	9.7	10.7	-9.7
								EDGE	1	13.6	9.7	10.7	0.0
FLOOR-33	35	20	18	9.8	49	VSLN	OK-N-F	DETACHED	1	13.6	7.7	10.7	-7.7
								EDGE	1	13.6	7.7	10.7	0.0

Table 4.1 Typical Design Session- Overall Building Configuration(continued).

Part-E: Evaluation Preferences Supplied by the Designer	
Feature	Value
Flexibility of rentable areas	100
Window line for rentable areas	100
Suitability for lateral structural system	100
High rentability at ground level	0
Travel distance from core to window-line	0
Clarity of circulation of rental areas	100
Daylight and view for core areas	0
Service connections at roof	0
Service connections at ground	0
General Energy Efficiency	100

Table 4.1 Typical Design Session- Overall Building Configuration(continued).

Part-F: Floor Evaluation Result (Not Sorted by Rank)				
Floor Plan Alternative	Storeys	Core-Type	Evaluation Value	Rank
FLOOR-1	10	CENTRAL	2200.0	1
FLOOR-2	12	CENTRAL	2100.0	2
FLOOR-3	15	CENTRAL	2100.0	2
FLOOR-4	19	CENTRAL	2000.0	3
FLOOR-5	23	ENDS	1400.0	5
FLOOR-6	28	ENDS	1400.0	5
FLOOR-8	13	CENTRAL	2100.0	2
FLOOR-9	16	CENTRAL	2100.0	2
FLOOR-10	21	CENTRAL	2000.0	3
FLOOR-11	25	ENDS	1400.0	5
FLOOR-12	31	ENDS	1400.0	5
FLOOR-14	14	CENTRAL	2200.0	1
FLOOR-15	17	CENTRAL	2100.0	2
FLOOR-16	23	CENTRAL	2000.0	3
FLOOR-17	24	DETACHED	1400.0	5
FLOOR-17	27	EDGE	1400.0	5
FLOOR-18	35	ENDS	1400.0	5
FLOOR-20	19	CENTRAL	2100.0	2
FLOOR-21	25	CENTRAL	2100.0	2
FLOOR-22	30	EDGE	1400.0	5
FLOOR-22	26	DETACHED	1400.0	5
FLOOR-23	39	ENDS	1400.0	5
FLOOR-25	22	CENTRAL	2200.0	1
FLOOR-26	28	CENTRAL	2100.0	2
FLOOR-27	35	EDGE	1400.0	5
FLOOR-27	30	DETACHED	1400.0	5
FLOOR-28	43	ENDS	1500.0	4
FLOOR-31	33	CENTRAL	2100.0	2
FLOOR-32	40	EDGE	1500.0	4
FLOOR-32	35	DETACHED	1500.0	4
FLOOR-33	49	EDGE	1400.0	5
FLOOR-33	43	DETACHED	1400.0	5

4. 4 Preliminary Cost Estimation

The comparison of design alternatives at the preliminary stage also involves cost comparison in addition to the qualitative design parameters used and listed in Fig. 4. 5. An accurate estimate of costs at the preliminary stage is difficult to obtain, with errors typically of up to forty percent (Neil 1982). As discussed in the next section, one of the most common methods of preliminary estimation is based on the building floor area and corresponding unit cost. Comparing different design alternatives having more or less the same gross floor area does not contribute to a conclusive decision. However some subtle variations in cost could be attributed to certain characteristics of the building. One such factor is the height of the building, which contributes definitely to an increase in the use of structural material and to some extent all other vertical elements in the building such as building exterior, service and HVAC ducts. Thus cost increases with height of building though gross floor area remains constant. Therefore, after qualitative evaluation at Level-II of the generation process, the building height is used as a parameter to calculate expected, relative structural costs for all the current alternatives. However it has to be mentioned that the relative structural cost based on height of the building may not be 'sensitive' enough when comparing buildings of smaller heights for which lateral load effects are not significant.

4. 4. 1 Different Methods of Cost Estimation

There are several methods for approximate or preliminary estimation. Some of them are based on cost per unit area, cost per unit volume, cost per unit surface area etc. Such methods give an order of magnitude estimate at the project inception stage. Unit-floor-area-based construction cost data are available for Canadian locations from sources such as Means (1996), Hanscomb (1995) and Lansdowne (1990). Different types of construction such as residential, commercial, office and industrial are included in such data. Adjustments for variations in the size of the project as well as geographical location of the project could be made with the data presented.

In general, cost estimates can be produced in increasing level of detail and accuracy by the following approaches:

- Unit area cost indices published by reputed organisations (as mentioned above);
- Interface the design system to material and assemblies cost data from database available in electronic form from such organisations as Means (1995) and then use the data to do cost estimation;
- Interface the design system to cost analysis software such as *Precision Estimating* (Timberline 1997) to perform a detailed cost estimate of the design alternatives based on detailed material and labour estimates and
- A comprehensive study, a life-cycle cost analysis of the alternatives so that other cost factors in addition to construction cost such as taxes, maintenance and energy are also taken into account.

The first method is solely based on the gross floor area of the building and serves the requirement for preliminary design giving an idea of the overall construction cost without need for design information on individual building subsystems. The second method is based on entire assemblies as opposed to individual material quantities. It requires the surface area or volume of each construction assembly (eg. floor type, beam-column type) for all building subsystems. The database of assemblies cost data is normally for 'in-place' items i.e. the labour cost being factored in. The third method above requires a reasonably good assessment of the quantities of material, equipment and labour. The fourth method requires, in addition to the construction cost a wider array of information such as energy costs, operating costs, municipal taxes, repair and maintenance costs at the location of the building.

4. 4. 2 Estimation Approach in Tall-D System

In the absence of detailed design and corresponding quantities, the building total cost is based on gross floor area of building using published unit cost data for office building construction (Hanscomb 1995). Such a method gives an approximate estimate without taking into consideration special or exceptional items that certain projects entail. In addition to site and owner conditions, the following factors affect the final building costs.

The height of building and construction material have varying degree of influence on the final cost. The building height is used as a handle to compare expected structural material costs. The depth of gravity system has an effect on the cost of other vertical building elements such as facade and the vertical structure (Colaco 1986). Therefore such a parameter is a useful indicator of relative cost of the gravity structural system. Tall-D estimates the relative cost of the different overall alternatives on the basis of building height. Upon selection of a particular layout, the system then generates the depth of alternative gravity systems, as part of the initial structural member sizing.

Another factor is the type of material - concrete, steel or composite. It has some influence on the total duration of the project, but more so on the start-up time. However the final cost variation due to different materials is not significant due to different cash flow patterns for steel and concrete buildings (Anthony 1985). Steel has a shorter start-up time, consequently results in higher cash-flow in the initial period as compared to concrete for which payment can be made as pouring of concrete is done. Taranath (1988) lists nine factors that have contributed to the cost-competitiveness of reinforced concrete for tall buildings, including forming techniques, concrete placing, fire-resistance, reduced floor-to-floor height, 410 MPa and 520 MPa yield strength (60 ksi and 70 ksi) mild steel, super-plastisizers etc. He also states that the slightly longer period that the concrete construction takes up will not significantly affect the overall cost. Glover (1991) in similar vein also states that the overall project times for both steel and concrete buildings are very similar. Therefore any variation in construction duration due to steel or concrete is not considered in the preliminary cost estimates. Bennett (1991), in a discussion on design using high-speed concrete, affirms that constraints in achieving speed and efficiency largely arise from the nature of designs. Therefore the effect of construction material on the initial cost of the project is not considered in Tall-D.

With the gross floor area being known after the generation of overall building configuration for all the alternatives, the approximate cost is estimated as follows. The Canadian national average cost per square metre for office buildings is used as the starting point. A first correction to that value is computed due to the location of the building, using cost indexes that are available for twelve major cities in Canada. Further correction due to the difference in size compared to the basic building is applied using a relationship that is valid for North America (Means 1996). Final correction for inflation is also applied to account for the date of construction.

After the estimation of the approximate total cost as above, the relative costs of the structure for the current building configuration alternatives is estimated. Fig. 4.7 is a typical relationship between the number of storeys and the weight of structural steel per square foot floor area in the building (Schueller 1986). The difference in material quantities for increasing number of storeys and resulting increase in lateral load effects can be observed from the figure. The relation shown in the figure is used to establish relative structural cost for building alternatives of different height. The primary aim in using this curve is to order the alternatives for expected structural cost. The unit quantities increase with increase in the number of floors both for steel and concrete buildings. Since the curve for total material for concrete would be similar in shape to that in Fig. 4.7 and considering that relative structure cost is sought rather than an accurate individual cost, the same relation as for steel is used for concrete as well.

The relative structural material quantities, having a direct bearing on structural costs, are presented as cost ratios with respect to the alternative having the least structural quantity. The comparison enables the designer to foresee any major variation in the structure cost among the current alternatives before proceeding to preliminary structural design. The initial cost is based only on the gross floor area of the building and is not based on whether steel or concrete is used as explained in section 4.4.2.

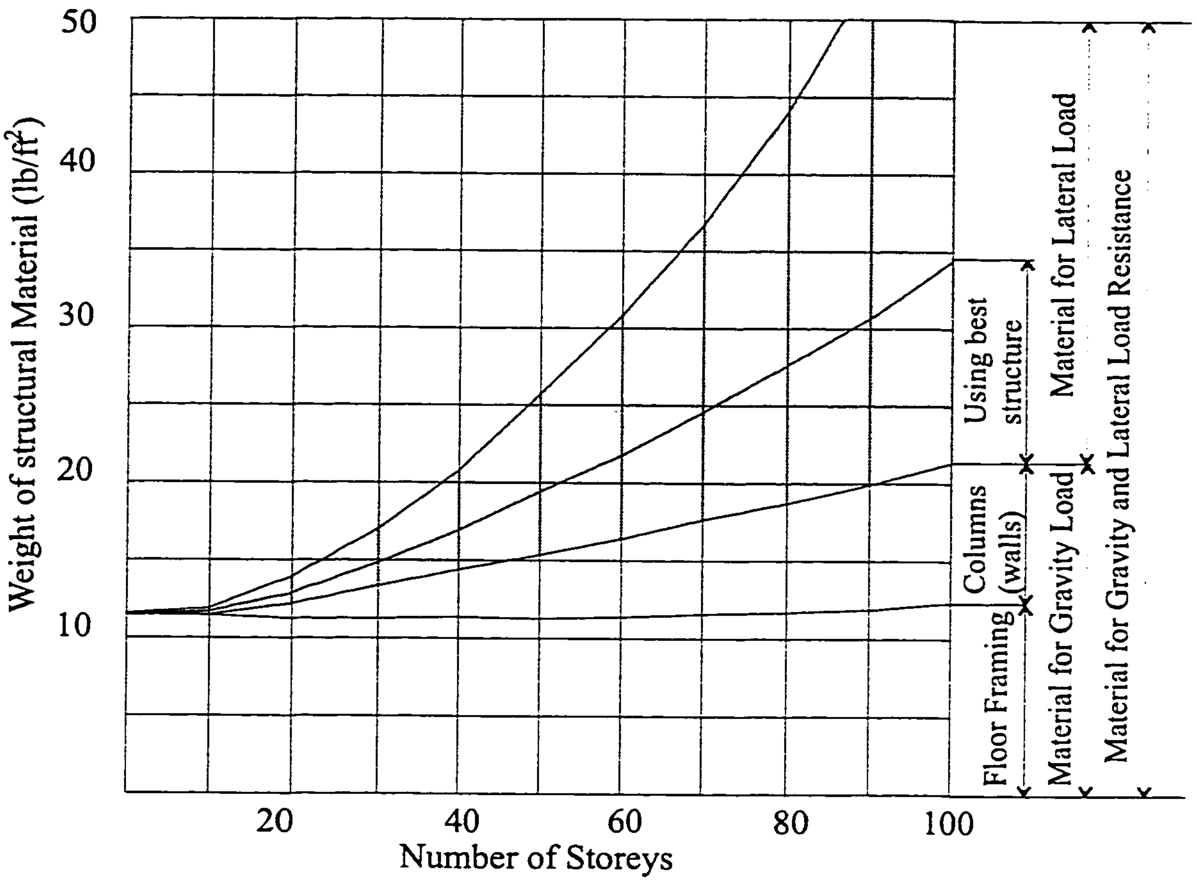


Figure 4.7 Variation of structural steel versus number of storeys (Schueller 1988).

4. 4. 3 Estimation for a Typical Building

The approximate cost estimation of the typical design presented in section 4.3.2 is shown here. The estimation is based on the unit (per square metre) cost of construction. Figure 4.8 shows an order of magnitude cost estimate for the total construction cost of the building. Figure 4.8 also shows a comparison of a different nature. The building configurations are evaluated prior to development of compatible structural systems to indicate possible variations in structural material quantities (and hence material cost). The comparison is based on the premise that increase in building height leads to increased unit structural materials in the building due to lateral load resistance requirements. As a general rule this is true although many instances may be sighted where the unit material consumption on a tall building is less than the material consumption on some other

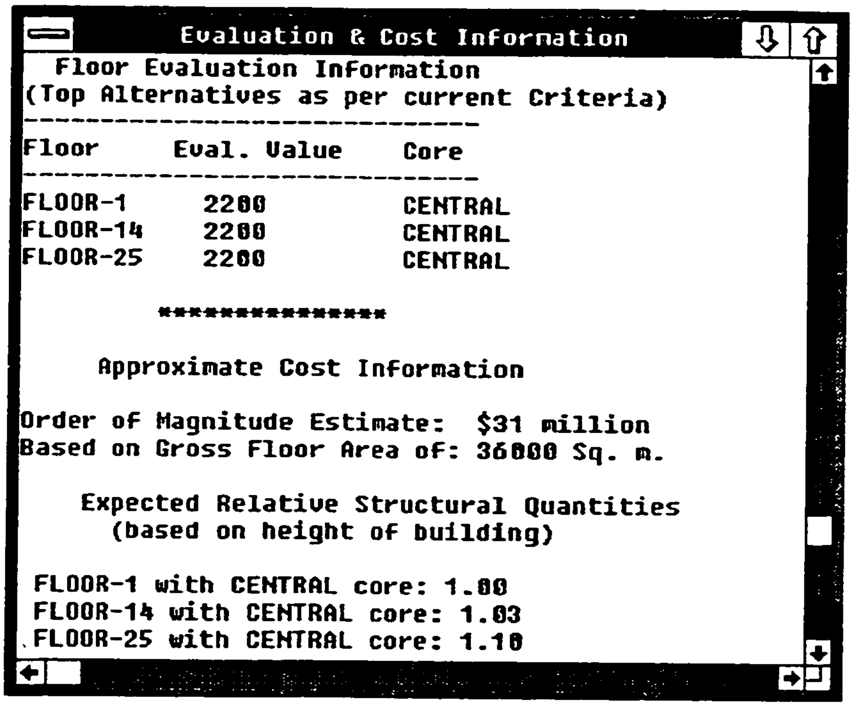


Figure 4.8 Order of magnitude estimate for example building.

building project with lesser number of storeys. Therefore the ranking presented in Fig. 4.8 for structural quantities is a guideline that designer may consider if a shorter building is desirable from other considerations as well. Since structural cost rarely exceeds thirty percent of the overall project cost, owners may not mind slightly increased material cost when aiming to build a tall building.

4. 5 Summary

The overall building configuration is representative of architectural design considerations where the building shape and massing are decided. Tall-D generates the floor outline alternatives and the number of stories along with the core location, considering typical zoning regulation for floor area ratio and building height limitations. A hierarchical generate-test procedure is implemented to prune alternatives in different stages using appropriate evaluation criteria. User can define preferences of design feature using a numeric factor. Order of magnitude cost estimates are made at this stage to confirm the budget limit specified by the designer. Historical cost indexes with correction for time, location and project size are applied in this cost estimating method.

CHAPTER 5

Structural Systems Configuration

It is not uncommon in practice, when major projects are being formulated, to consider many alternative solutions. Office buildings with large rentable space are major investment decisions and, as such, thorough analysis of not only the commercial aspects, but analysis of the engineering aspects as well is called for. There are documented cases of investigation of alternative structural configurations (Colaco 1986, Taranath 1988, Khan and Iyengar 1973, Iyengar 1977). Small projects by nature are not subject to extensive evaluation of alternatives either due to constraints on design resources or because the solution is often apparent and less penalizing economically.

Given the fact that the generation of alternative designs is a common practice inherent to the design process in the context of multistorey office buildings, Tall-D attempts to simulate such a process. The structural module in Tall-D generates and evaluates alternative structural schemes that are compatible with the overall configuration of the building. Vertical and horizontal structural systems are both considered. Generic systems are initially selected, followed by the generation of the lateral load-resisting system (LLS). Feasibility of the combination of gravity load-resisting system (GLS) with LLS is verified and compatible alternative gravity systems are generated. The structural module also enables the designer to examine the geometry of the generated configurations in plan and elevation as well as in 3D wireframe by the use of an interface to AutoCad.

The following terms are often used interchangeably: lateral load-resisting system(LLS) and vertical system. However, the LLS refers only to that part which, in addition to gravity load, also participates in wind and seismic resistance, whereas the vertical system also encompasses those columns that do not contribute to wind or seismic resistance. Similarly gravity load-resisting system and floor system are often used synonymously. This chapter begins with a discussion of the lateral-load resisting system, the different types, how Tall-D configures some of these LLS's and issues related to design integration. Geometric configuration, or in other words the layout of columns performed by Tall-D, is also presented. Then issues of the synthesis of the lateral system with the gravity system are presented. The structural system alternatives generated for the example building reported in section 4.3.2 are included in the discussion. The representation of the structural system and components as physical objects in the knowledge-base is finally discussed in section 3.3.

The overall building configuration consists of details of the structural system such as the type of lateral and gravity systems, construction materials, geometric and location information of the different LLS components. Simplified design of structural systems is discussed (i.e. simplified analysis and member sizing) in section 5.3.

Examination of typical applications reviewed in section 2.5.2 shows that suitable analysis and design techniques (in this case related to structural design) are required to complement the production rules in a KBS environment. Often, the solution adopted was to interface with detailed analysis programs (Kumar and Topping 1988). In the context of preliminary design such an interface is not practical for the development of structural member sizes of all the different alternative configurations. A lot of input preparation would be required for each alternative, whereas many alternatives demand only quick design. Tall-D demonstrates the use of approximate design methods for the many different structural system alternatives it generates.

Approximate methods of structural analysis and design not only preclude the use of conventional analysis programs at the preliminary design stage, but in most cases are commensurate with the degree of accuracy expected at the preliminary design stage. The effort to build appropriate interfaces to such external programs is then saved. The approximate methods are also more flexible, in modelling decisions for multistorey building structures, than the general-purpose analysis programs. Furthermore, hand computations can be performed to verify quickly the results produced by the Tall-D system. Some of the issues with regards to the use of approximate structural analysis and design methods in a KBS environment are also addressed in section 5.3.

5. 1 Lateral Load-Resisting System (LLS)

The selection of a LLS for a multistorey building has a significant impact on the construction costs as well as on the functionality of the occupant space. The suitability of different LLS with respect to floor layout flexibility, height and slenderness characteristics of the building is taken into account. Alternative lateral load-resisting systems that can be assembled include rigid frames, braced frames, frame-shearwall system and tubular frames. Other important considerations include compatibility between the lateral and gravity load-resisting systems as well as the use of steel and concrete. Though Tall-D does not, at the moment, perform approximate member sizing for composite construction, it identifies the situations where composite construction is worth exploring.

Following the evaluation and ranking of overall configurations in the previous module, the structural configuration module is ready to proceed, having prior relevant information such as the number of storeys, slenderness, plan dimensions, core size and location. Generic lateral load-resisting systems are first identified as compatible with the overall building configuration under consideration (Coull and Stafford Smith 1991, Taranath 1988, Council on Tall Buildings 1979).

Factors such as the height of the building, the shape of the floor plan, location of the core, construction material and clear span requirements determine the initial layout of the lateral structural system. The layout information generated by Tall-D includes column spacing, hence the number of bays in each direction in the case of rigid-frames, braced-frames or shearwall-frames. Where the space between the core and the window-line is clear-spanned, the rigid-frames on the window lines are used to significantly contribute to lateral load resistance by close spacing of columns as well as deep spandrel beams. Additional gravity columns (that take only vertical loads) as required in some situations are also located on the structural layout plan. The columns inside the core are also laid out tentatively since the precise layout of the elevators, shafts and service areas are not determined by Tall-D. In the case of a frame-shearwall, the number, location and direction of the principal walls are determined in addition to column details on the perimeter. Principal walls here are defined as those walls that are designed to partake in lateral load resistance as opposed to non-load bearing or gravity load bearing walls and shafts. Incidentally, gravity loads on such principal walls contribute to their lateral load performance by increasing the threshold of onset of tension in shearwalls.

In braced frame buildings decisions are concerned with the type of bracing, location in the perimeter or the core with consideration of the significant effects the braces will have on the internal flexibility of the space as well as the visual effect if it were on the exterior. The latter issue however is only considered in Tall-D by allowing the user to select or ignore schemes that have braced frames on the perimeter.

In the case of a tubular system, decisions are related to the spacing of perimeter columns (Taranath 1988, Spires & Arora 1990), the type of core, either with shearwalls or gravity columns, as well as intermediate gravity columns if clear span between of the core and the window line is too long. Other decisions are concerned with the type of connections between beams and shearwalls (hinged or rigid), considering the need to accommodate differential axial deformations between the exterior columns (exposed to outdoor temperature change) and the interior concrete core (shortening due to creep and shrinkage).

The characteristics of the lateral system affect the selection of the gravity system. For example, column and beam optimum relative proportions maximise efficient material use. Column spacing determines the main spans and hence the type and depth of floor system. Alternatively as in some instance when a particular floor system is retained to begin with, the LLS characteristics influence the practicality of the chosen floor system. The vertical system also affects the internal core configuration if the load-resisting part is located fully or partly in the core. If flat-slab or flat plate is considered for the gravity system due to overall lower floor-to-floor height and flexibility in integrating horizontal service distribution, then lateral stability has to be achieved by closely spaced columns inside or on the perimeter of the building or by an internal shear-core. Shear-core here refers to a set of shear walls or a set of steel braced frames acting as the principal lateral load resisting elements. Thus generic gravity load-resisting systems that are compatible with each type of lateral system are identified for further processing by the Tall-D system (Council on Tall Buildings 1980, Taranath 1988, Allen and Iano 1989).

Finally the internal subdivision of the core is composed either of a set of shearwalls or a grid of columns that fit within the core. Since the core configuration is largely dependent on the elevator banks, the air-conditioning supply and return air ducts and other service ducts, service area location and fire exit location, all of which fall typically in the realm of detailed design, these factors are not addressed in detail by Tall-D. Hence a tentative grid or arrangement of shearwalls is adopted as core configuration by Tall-D as only adequate for preliminary design.

The location of the mechanical level(s) depends on the number of storeys and may be used to locate outrigger trusses when additional stiffness in the lateral system is called for.

5. 1. 1 Structural Schemes and Integration Issues

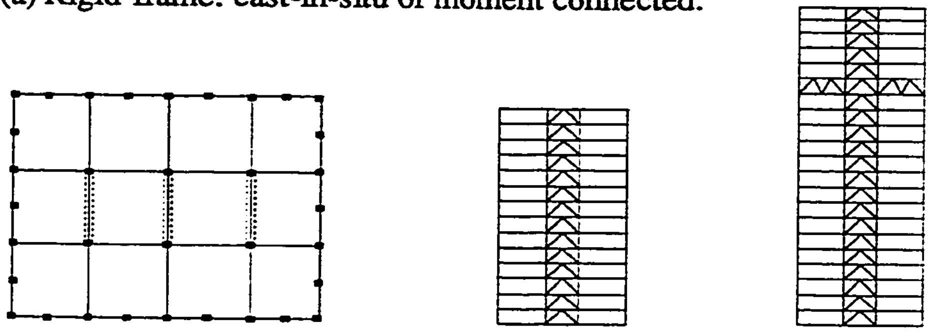
Following is a list of generic types of lateral load-resisting structural systems that are used in multistorey buildings. The different structure types are categorised as per the principal lateral load-resisting system along with variations in each category.

- Rigid frame (Figure 5.1a)
    - i) Rigid frames in the whole building (all moment connections).
    - ii) Rigid frames only on the perimeter of the building.

Regular grid

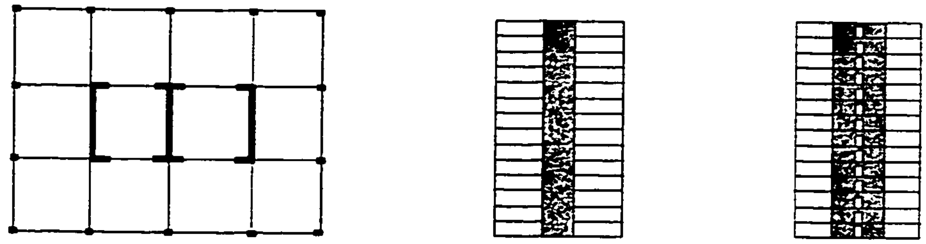
Close Perimeter column spacing

(a) Rigid-frame: cast-in-situ or moment connected.



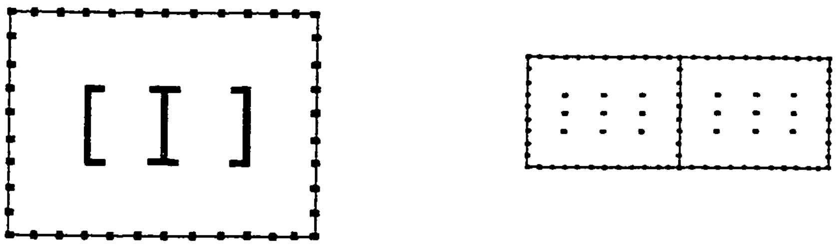
Braced Frame

(b) Braced-frame: braces located in core or on perimeter; stiffening 'belt' truss.



Shearwall scheme

(c) Shearwall scheme: independent action or frame interactive systems.



(d) Tubular schemes: close column spacing on the perimeter with deep spandrel beams.

Figure 5.1 Generic types of lateral load-resisting structural systems for multistorey buildings.

- Braced frame (Figure 5.1b)
  - i) Braced frames in the core of the building.
  - ii) Braced frames on the perimeter of the building; rigid frame interactive systems.
  - iii) Braced frame with outrigger: braced frames in the core of the building with hat/belt truss around the building.

- Frame-Shearwall (Figure 5.1c)
  - i) Only shearwalls participate in lateral load resistance.
  - ii) Both shearwalls and frame participate in lateral load resistance.

- Framed Tube (Figure 5.1d)
  - i) Tube with closely spaced columns
  - ii) Tube with widely spaced columns.

Tall-D can configure the geometry for most of the above structure types, except the outrigger in Figure 5.1(b). Numerous variations, adaptations and combinations of these systems have been used. Tall-D can also perform the initial structural member sizing for most with the exception of the outrigger system. The vertical structural system in the outrigger scheme can be designed based on a routine frame analysis and applying a reduction to the resulting moments accounting for the effect of the outrigger truss. The introduction of the outrigger truss reduces the wind moments on the braced frame system by as much as two thirds at the base of the building and even more close to the level of the outrigger (Cheong-Siat-Moy 1991).

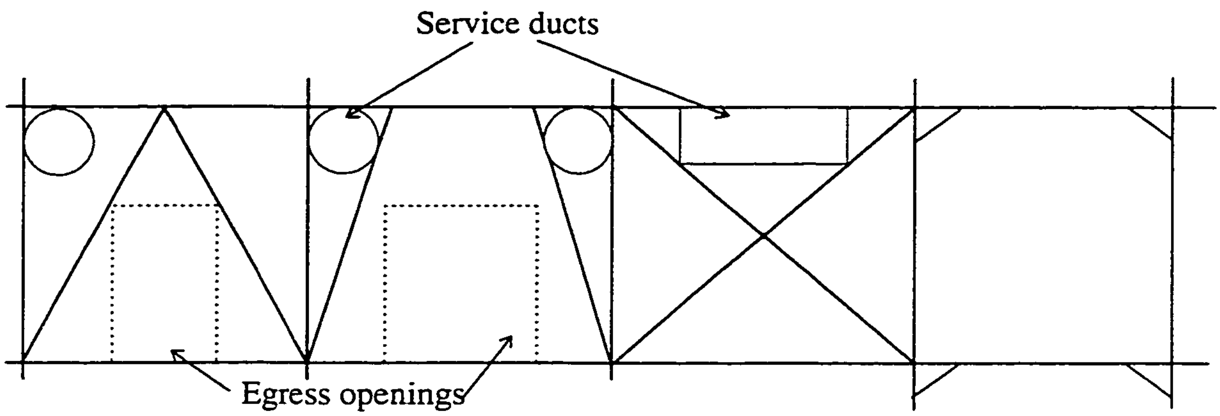
Integration Issues

The lateral load-resisting system in multistorey buildings constitutes a vital element. It is therefore necessary to integrate the structure with the rest the building subsystems. Some of the issues that Tall-D addresses in this regard are presented here by means of specific examples.

Rigid Frame: Rigid frames are quite desirable from the point of view of integration in buildings since they have rectangular portal openings. With this type of LLS it is easy to work ducts around the plan of a building. Rigid frames offer a great degree of flexibility to locate openings for doors and windows. Visually, they do not pose any hurdle to architectural expression. Rigid frames are also well suited to increase the flexibility of the occupant space since large spans can be obtained by making the perimeter frames sufficiently stiff. Indeed lateral loads can be carried by much stiffer frames on the perimeter of the building by means of reduced column spacing and deeper beams.

Braced Frame: Braced frames are efficient structural systems. By virtue of resisting external loads predominantly with axial forces, they relieve the beams and columns of substantial moments due to lateral loads (Cheong-Siat-Moy 1991). Floor beams of smaller depth can thus be used, thereby reducing floor to floor height. It also follows that uniform floor depth over the height of the building leads to economy by way of repetition. In most instances the braced frames are placed in the core, reducing obstructions for openings as well as reducing visual impact. Even when it is necessary to place the braced frames in occupant areas with a very small core, a judicious mix of placement of openings and different bracing types (eg. knee bracing) can easily solve the problem of occupant flexibility (Figure 5.2). Braces can become part of the architectural expression of a building when present on the exterior of the building which also increases the flexibility of the occupant space.

Braced frame with outrigger: Outrigger trusses at one or more levels along with perimeter trusses at the same storey levels are used to mobilize the columns on the perimeter more actively to improve structural efficiency against lateral load (Stafford Smith and Coull 1991, Kowalczyk et al. 1995). Since these storey-deep outrigger trusses run across the building, they are not feasible on floors of normal occupancy. However, in the case of large multistorey buildings, floors that are entirely allocated for mechanical equipment can be used to place these outriggers as well as the perimeter trusses. Thus, integration between the mechanical space and structure is achieved to a high degree. It also satisfies the design rationale of an efficient lateral load system. Also in situations where braced frames alone cannot meet the drift or strength requirements, outrigger truss at the mechanical level can be introduced to meet such requirements.



K-brace
Adequate openings and least bending in floor beam

Knee brace
Large openings and more bending in floor beam

X brace
Least flexible in terms of openings more bending in floor beam

Rigid frame
Maximum opening and more bending in floor beam

Figure 5.2 Different types of LLS bracing in relation to service ducts and egress (adapted from Eligator et al. 1990)

Frame-Shearwall: Where there is a consolidated core (examples: CENTRAL, EDGE and END configurations in Fig.4.4), as in many multi-storeyed buildings, shearwalls represent an efficient way of resisting lateral loads. In buildings taller than twenty storeys, the additional contribution of the beams and columns to lateral load resistance can be taken advantage of. By considering the interaction between the frame and shearwalls, a more efficient LLS can be designed than using walls alone.

Framed Tube: Tubular structural behaviour is achieved in multistorey buildings primarily through close spacing of columns as well as deep beams around the perimeter of the building. The structure then responds like a prismatic tube with openings against lateral loads. It is feasible to induce such a response only if the concentric frames making up the tube are sufficiently stiff through close column spacing and deep spandrel beams. Tubular structural configurations are also very efficient LLS with most of the wind bracing at the perimeter of the building, which leaves the interior with fewer, smaller gravity columns. The result is maximum flexibility in occupant space, achieving high degree of integration between structure and space planning considerations. Furthermore, tubular configurations are not restricted to rectilinear plans nor prismatic buildings. Using bundled tubes, not only larger plan footprints can be achieved, but also individual tubes can terminate at different levels along the height of the building, thus creating a varied floor plan.

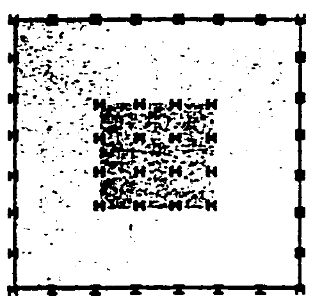
5. 1. 2 Geometric Configuration of Lateral Load-Resisting Structural Schemes

There is a variety of possible configurations in which the building structure can be arranged. In addition to the above listed types of LLS, the structural designer must also consider their location in the building as well as the location of the core and the possibility that more than one type of structure is feasible in a given design context: these possibilities thus give rise to a problem of geometric reasoning. Tall-D demonstrates the feasibility of generating these geometric configurations for most of the layouts.

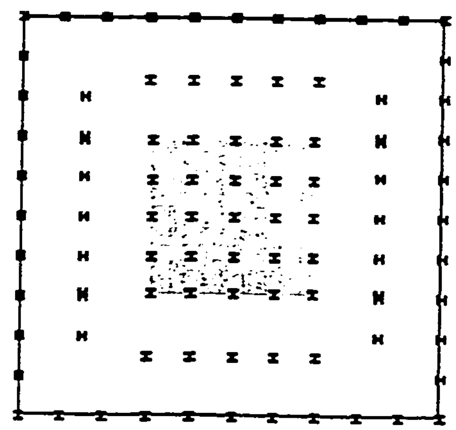
The Tall-D knowledge-base uses the 'Column-Layout' knowledge-representation frame to generate instances of the possible column, shearwall and brace locations. Figure 5.3 shows some of the generic column layout alternatives in Tall-D. It may be noted that some of the layouts are better suited to steel than concrete. For example close spaced columns in one direction and long spans in the other are more suited for steel than concrete due to the nature of steel floor framing (Glover 1991). The generic layout types are given self descriptive names *Grid-2D, Grid-Aligned-to-Core, As-per-Shearwalls* or *Perimeter-Based*, for purposes of clearly defining the types of possible column layouts as well as implementation in the Tall-D system. Tall-D implementation uses this formalism in writing the rules in the knowledge base. Appendix C lists the names of the many rules that configure column layouts for different structure types. See pages C-7 through C-15 for this list and pages C-33 through C-36 for a list of Lisp functions many of which are related to column layouts and structure configuration.

In the *Grid-2D* column layout the columns are located at all grid points in plan resulting in uniform bays and uniform aisles, typically encountered in the lower end of high rise buildings . The grid lines could be spaced equally in both directions or otherwise

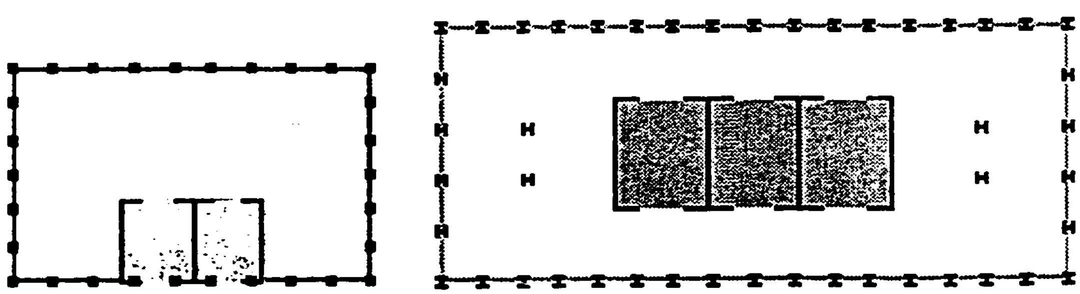
(a)-(b) Grid-2D



(c) Perimeter-Based



(d) Grid-Aligned-to-Core



(e)-(f) As-Per-Shearwalls

Figure 5.3 Generic column layout alternatives from Tall-D.

(Fig. 5.3 a and b). These layouts are used for rigid-framed buildings and semi-rigid (steel) buildings of moderate height (i.e. up to fifteen storeys).

In the *Perimeter-Based* column layout, the close spacing of columns is given consideration. This layout is preferred when clear-span is a feasibility or high on the designers preference (Fig. 5.3 c). There may be a set of columns in the core.

In the *Grid-Aligned-to-Core* column layout, an intermediate row of columns is introduced in one or both directions based on the core to window-line distance as well as designer initial preference (Fig. 5.3 d) . Due to the possibility of LLS being placed in the core, close spacing of beams as well as deep spandrel beams are not primary issue in this type of layout. Since this layout emphasizes columns in core and elsewhere, it is most suited to configure steel structural systems.

The *As-per-Shearwalls* column layout is used where shearwalls are the primary LLS. Since shearwalls are located in the cores (eg. CENTRAL, EDGE or END core etc.) this type of column layout takes into account the locations that are not available for placing columns due to the presence of the shearwalls. This layout is suited for configuring frame-shearwall interaction systems as well as tubular systems.

Though the above categories of column layouts are not associated with the material of construction, the rule-base incorporates selection mechanism to ensure appropriate layout types are used during structural configuration. In some instances a specific case of one layout could eventually look like another layout type. An example is when a *Grid-Aligned-to-Core* layout without intermediate row of columns looks like *Perimeter-Based* layout. However, the *Perimeter-Based* layout would have alternatives with closer column spacing on the perimeter than the earlier one.

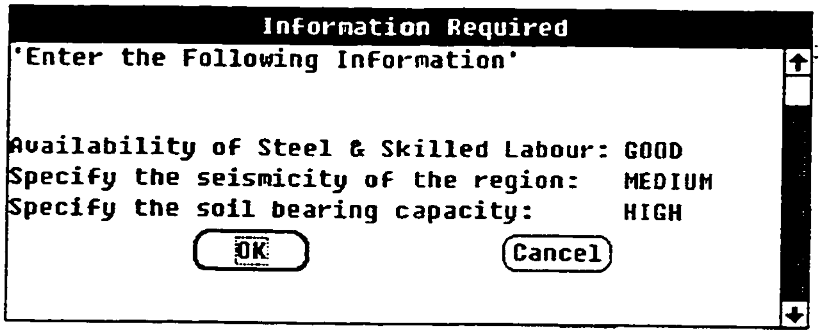
These categories thus serve to loosely formalise the most commonly encountered structural system layouts and as a reference point for the discussion as well as implementation.

The geometric description of bay spacing, perimeter column spacing, location and thickness of shear walls, bracing configuration etc. for the LLS is produced as part of the different instances of the lateral system. It is also used for display as well as in modelling for approximate analysis where available.

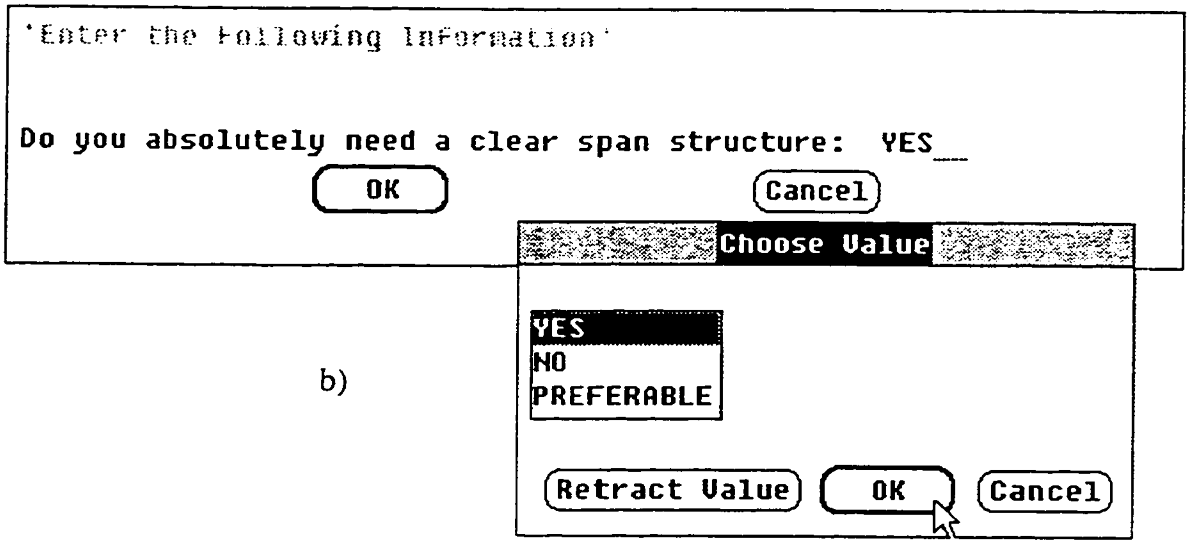
5. 1. 3 Example: Configuration of a Typical LLS

For purposes of illustration, we continue with the typical design session initiated in section 4.3.2 and generate LLS alternatives to be compatible with overall configurations Layout#25, Layout#14 and Layout#1 of Figure 4.6. These three layouts were evaluated with highest ratings. The input conditions set by the designer for the generation of structural alternatives are shown in Figures 5.4(a) and 5.4(b). The designer is asked for such information as the availability of steel, seismic conditions, soil conditions as well as requirement of clear span. Soil condition is required for feasibility of heavier concrete structure. The names of all the different structural schemes generated for Layout#25 are shown in a selection list in Figure 5.4(c), the details of which are discussed further.

The overall configuration represented by Layout#25 has 22 storeys, with plan dimensions of 40 x 40m and a central core of 13.3 x 13.3m. Figures 5.5(a), 5.5(b) and 5.5(c) show in plan and elevation the frame-shearwall alternatives. The plan shows the location of the columns and shearwalls and the elevation shows the rigid frame that is on the facade of the building as well as a transverse frame with a shearwall that passes through the core. The shearwall may in practice have openings, which is a decision more appropriate for the detailed design stage. The space between the perimeter and the core is column-free so as to maximise occupant flexibility and rentability. There is associated explanatory text with each screen presented in the figure indicating the geometry, orientation as well as the instance names of the structural scheme and plane frames. Here, it may be noted that the numbers that are part of the instance names are not indicative of the total number of alternatives, but are system-generated so as to be unique. The difference between the three shearwall alternatives is the column spacing on the perimeter. The core has three shearwalls, each with flanges that act as shearwalls (with openings) a)



b)



c)

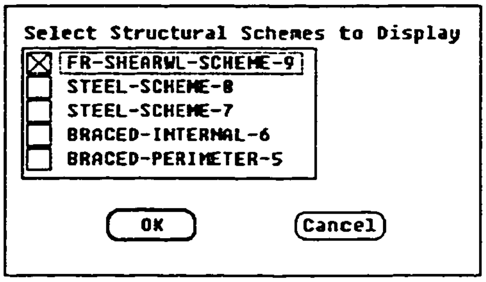
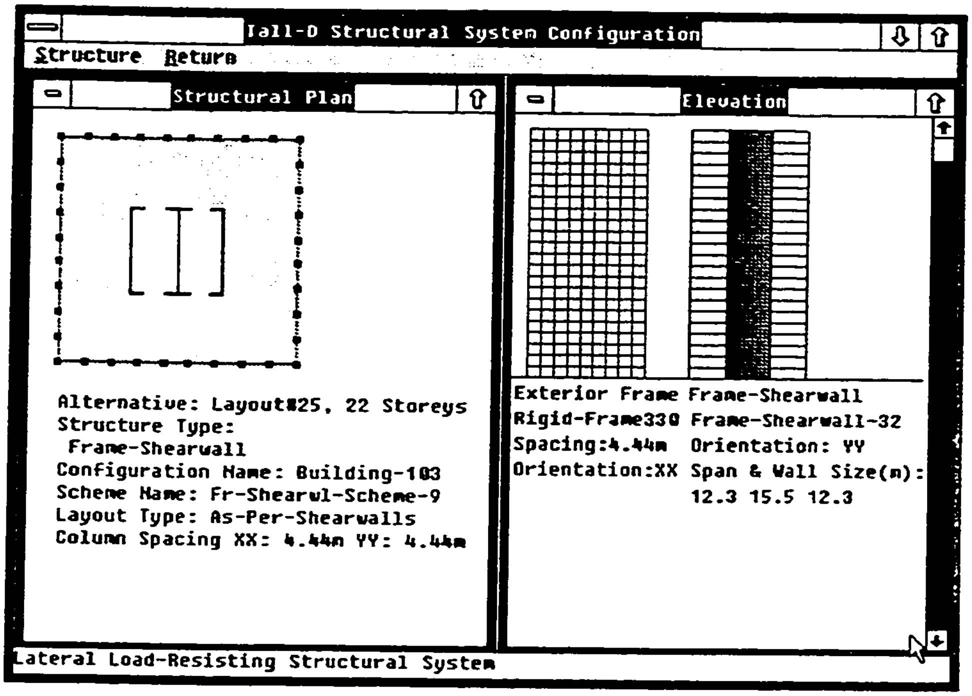
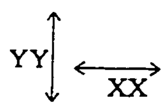


Figure 5.4 (a) Design constraints set by designer before structural alternatives are generated. (b) Clear span option selected by the designer. (c) List of instance names of alternative structural schemes generated for Layout#25.

(a)





(b)

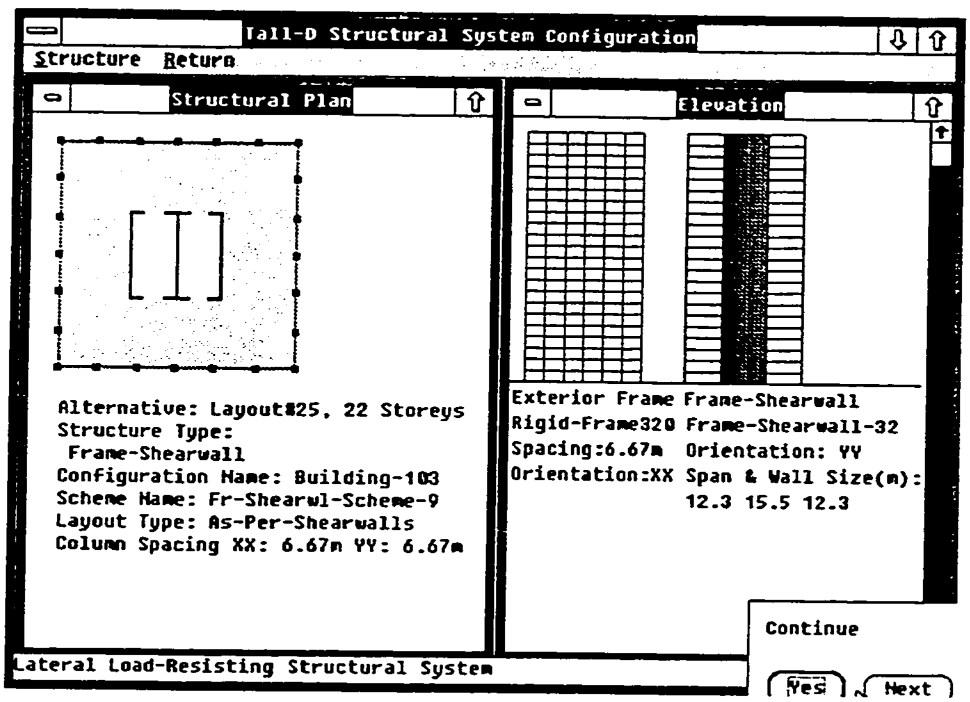


Figure 5.5 Frame-shearwall structural system alternatives for Layout#25: Perimeter column spacing: a) 4.44m on both sides; b) 6.67m on both sides.

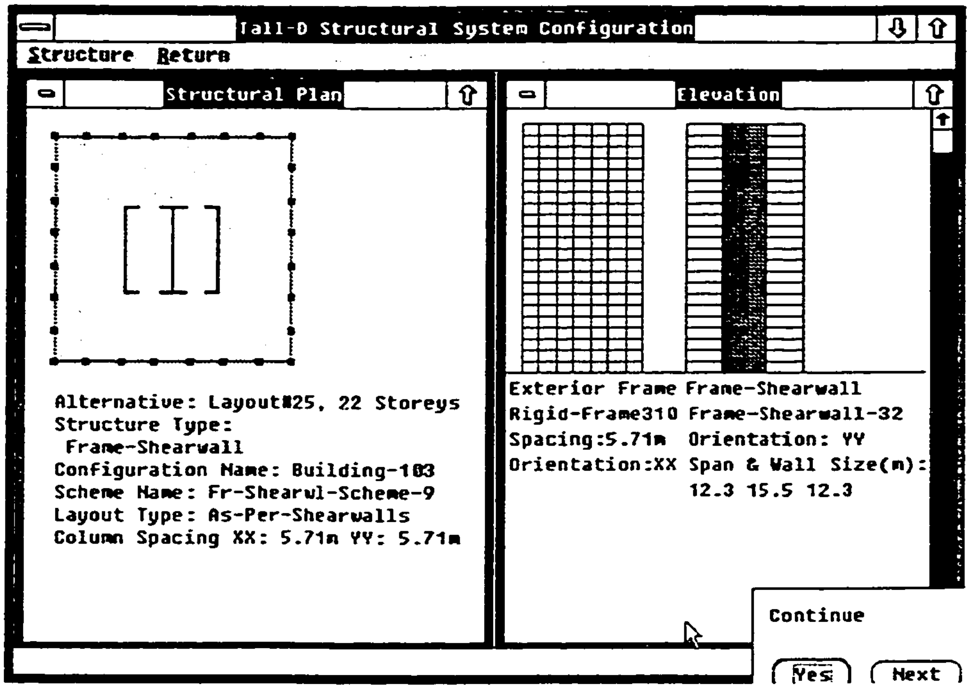


Figure 5.5(c) Frame-shearwall structural system alternative for Layout#25: Perimeter column spacing 5.71m.

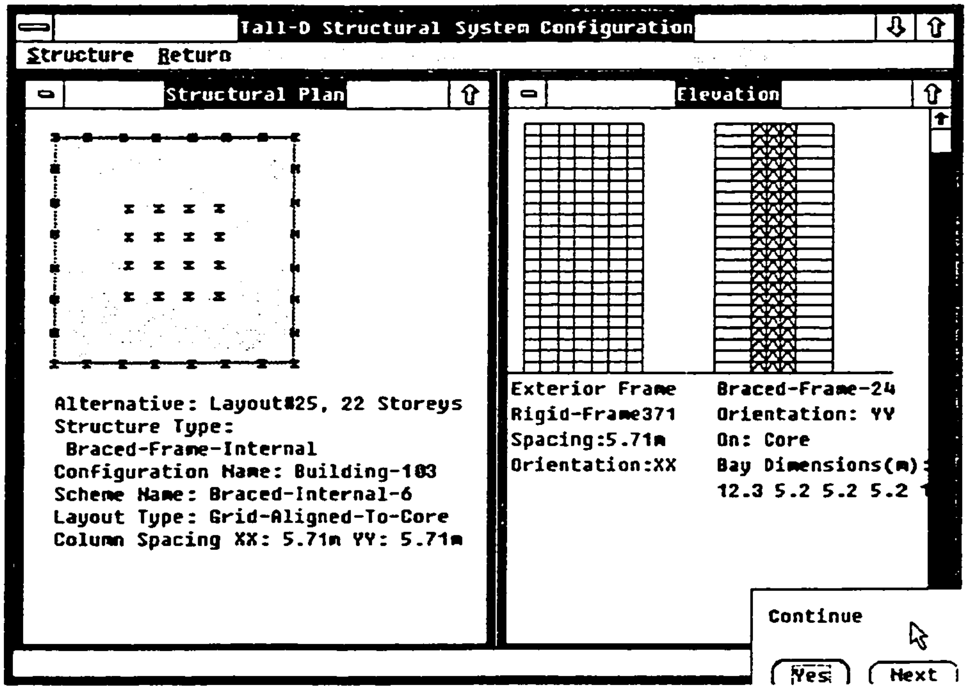


Figure 5.5(d) Braced frame structural alternative for Layout#25: Braces located in the core. Perimeter column spacing 5.71m.

in the other direction. Additional walls in that direction (i.e. XX) for additional lateral support, can be placed in the core as required at the detailed design stage.

The subsequent Figures 5.5(d)-(g) show braced frame alternatives. The first two braced frame alternatives have the braces placed in the core. The difference between the first three braced frames is the type of column layout. The first one (d) is a clear-span alternative, i.e. column-free core to window line distance all around the core. The second (e) of these is a Grid-2D alternative with unequal bay and aisle sizes, i.e. small bays, large aisles. Large aisles are generated, even though the designer may have specified a clear-span requirement. In fact Tall-D generates large span alternatives along with clear-span alternatives when clear span is requested. Bracing, by resisting the lateral loads by axial forces in members, significantly reduces the lateral load moments in beam members, resulting in acceptable beam depth in the other bays of the frames that do not have braces. Bracing also permits uniform floor system depth due to the fact that the beam depth is now controlled by gravity loads on beams that are uniform along the height of the building.

The last two braced frame alternatives in Figures 5.5(f) and 5.5(g) have the braces placed in the perimeter frames. This arrangement eliminates the braces from inside the building, which may be desirable from the occupant flexibility point of view, as in buildings of moderate size where the core is much smaller than in large slender buildings. Structurally this also creates an efficient lateral load-resisting system, and in slender buildings converges to brace-framed tubular system if the braces cross the entire face of the building on each side. Exterior braces have also been used for architectural expression in some instances. Where such an exposure is not desirable, this alternative is not feasible.

The final structural scheme alternatives considered for Layout#25 are steel rigid frames. With the number of storeys being 22, the rigid joints in steel frames become justifiable, as they enable the mobilization of the full potential of steel frames to resist lateral loads. It has the added advantage of brace-free bays and aisles, preferred by the designers against braced frames in the building. Alternative column layouts for this steel

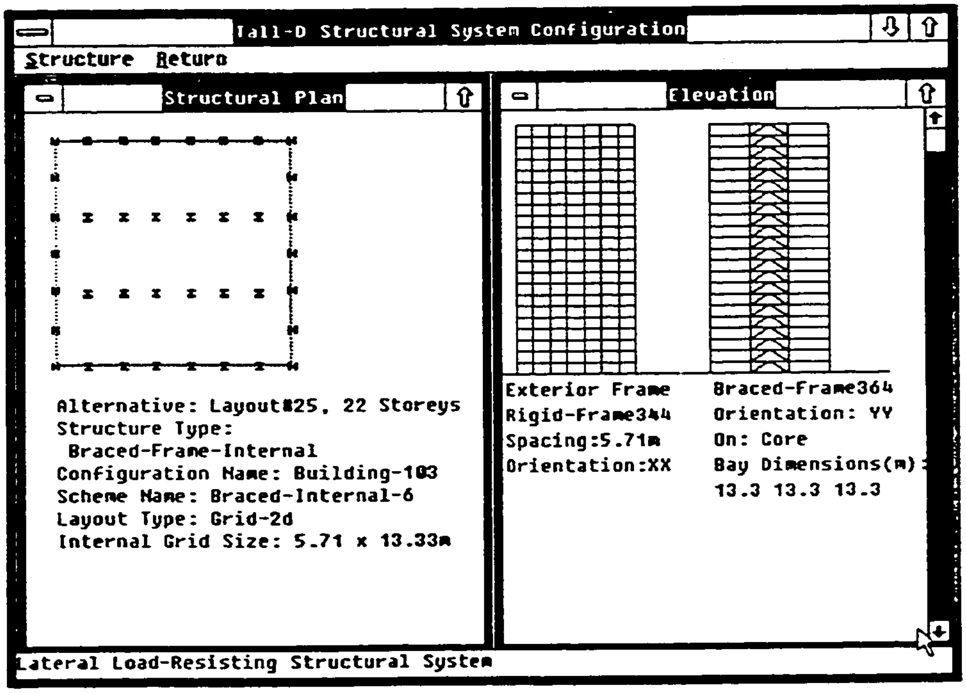


Figure 5.5(e) Braced frame structural alternative for Layout#25: Braces located in the core. Column grid 5.71x13.33m.

scheme are presented in Figures 5.5(h) through 5.5(n). The first two are alternatives (h and i) with close column spacing on the perimeter, so as to relieve the occupant space of columns, to result in a clear span from core to window-line. The next two alternatives (j and k) are of type grid-aligned-to-core. Perimeter-based and Grid-aligned-to-core types sometimes look similar. Alternatives in Figure 5.5(i) and 5.5(k) converge to be the same. However in the grid-aligned-to-core layout of Fig. 5.5(k), the moment connections are not restricted to the perimeter alone as in the case of the perimeter-based layout of Fig. 5.5(i). The columns in the core are set to a practical grid size that fits the core. As more details of the core become available the core column layout may be subject to modifications. However such details could be considered after the preliminary design stage.

The subsequent two alternatives in Figures 5.5(l) and 5.5(m) show a column layout type Grid-2D with unequal bay and aisle sizes, i.e. one with 5.71m bay size and the other with 4.44m bay size. In steel buildings the linear arrangement of beams that results is advantageous to generate economical floor systems. The final alternative shown in Figure 5.5(n) is one of the additional alternatives that is generated if the clear span requirement

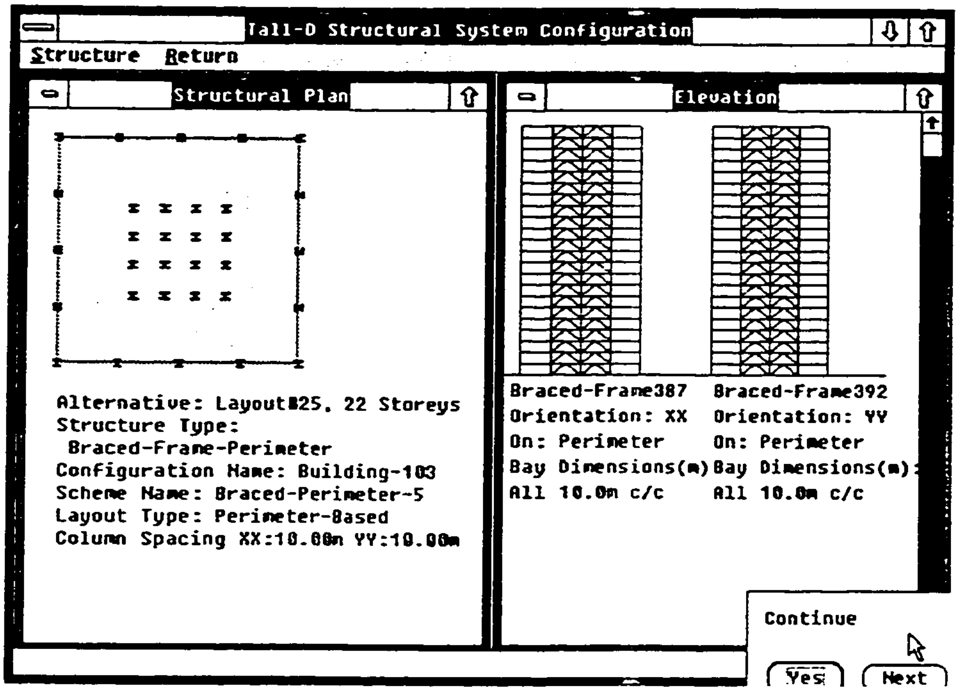


Figure 5.5(f) Braced frame structural alternative for Layout#25: Braces located on the perimeter. Perimeter column spacing 10m.

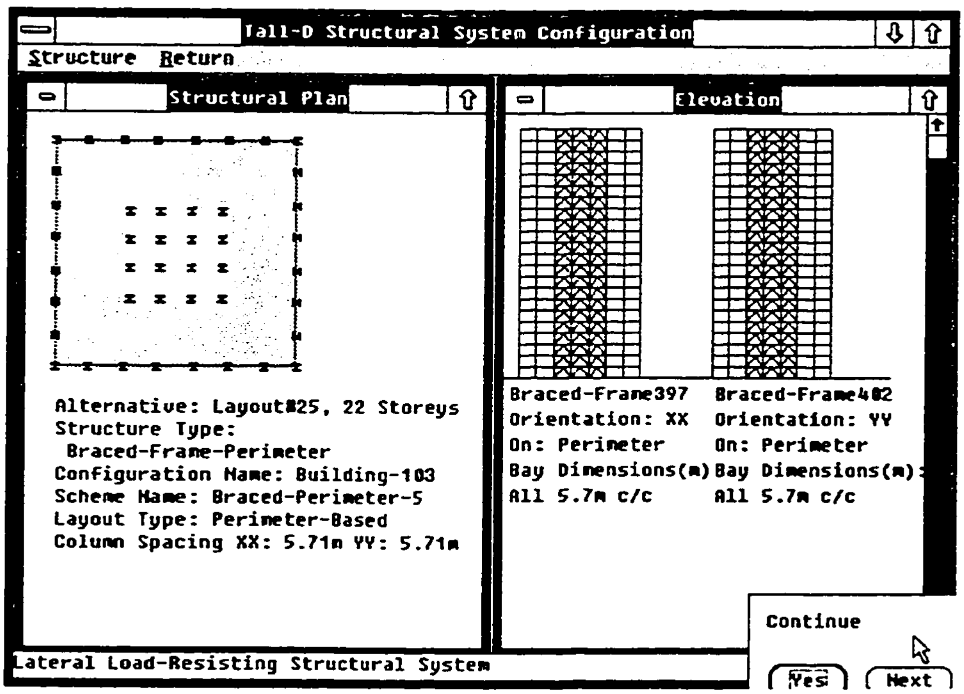


Figure 5.5(g) Braced frame structural alternative for Layout#25: Braces located on the perimeter. Perimeter column spacing 5.71m.

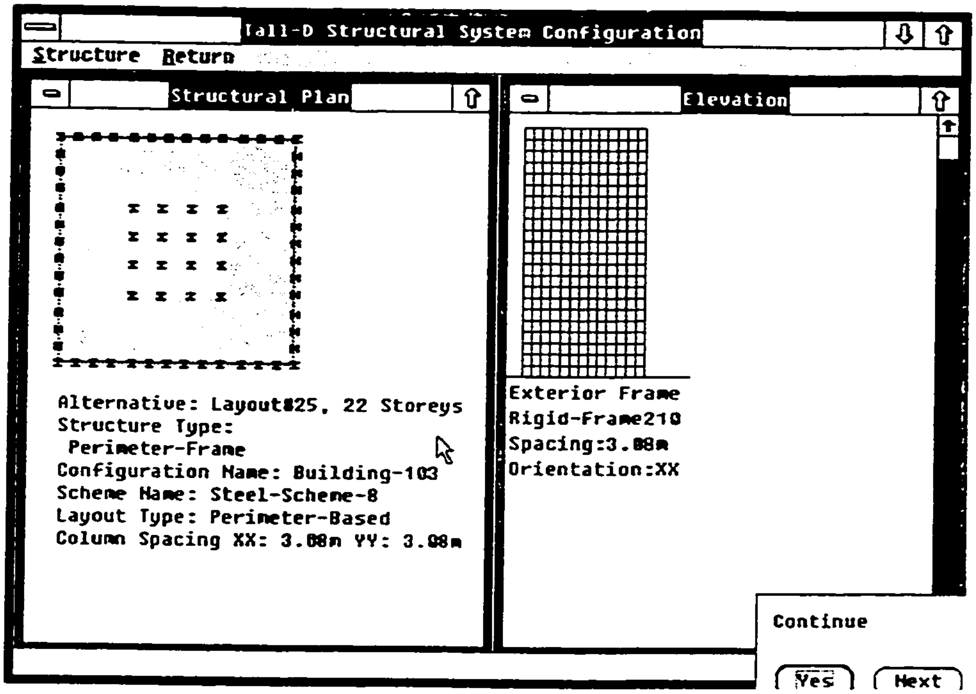


Figure 5.5(h) Perimeter-based rigid frame structural system alternative for Layout#25: Perimeter column spacing 3.08m.

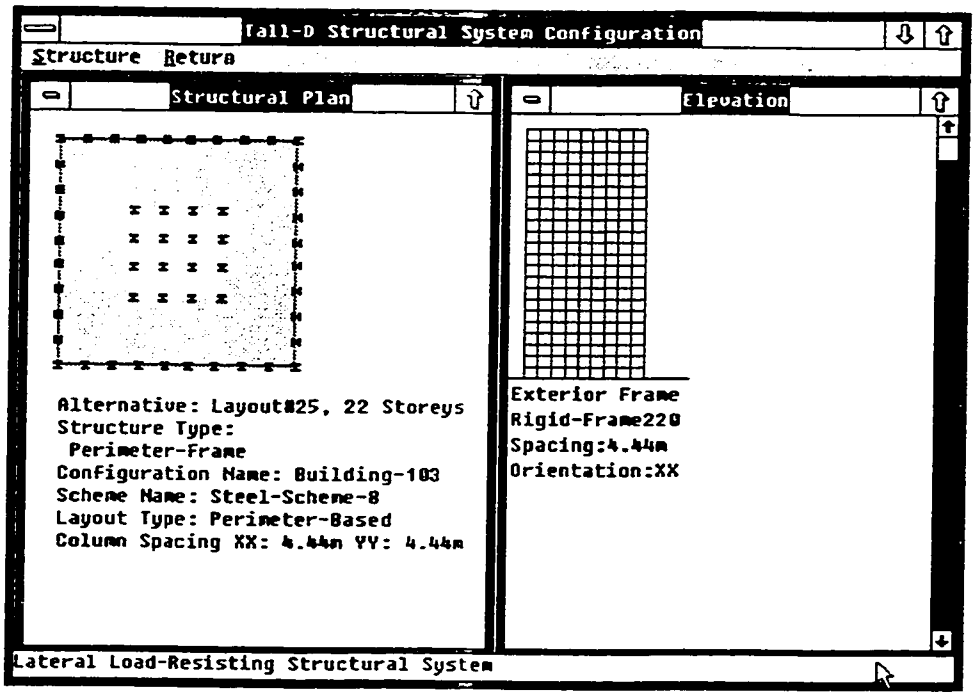


Figure 5.5(i) Perimeter-based rigid frame structural system alternative for Layout#25: Perimeter column spacing 4.44m.

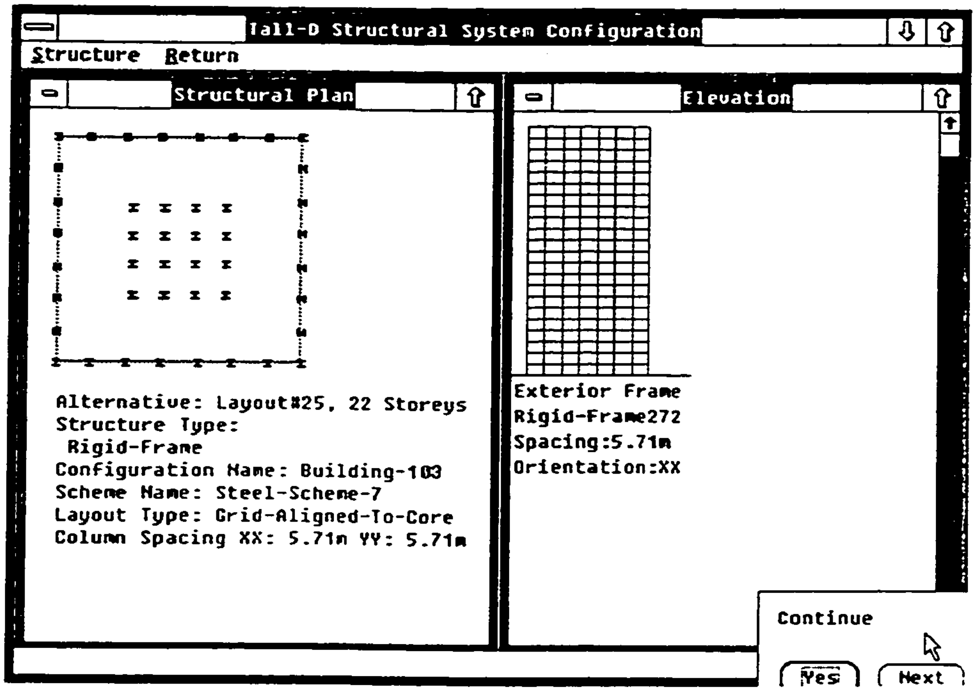


Figure 5.5(j) Rigid frame structural system alternative for Layout#25: Perimeter column spacing 5.71m.

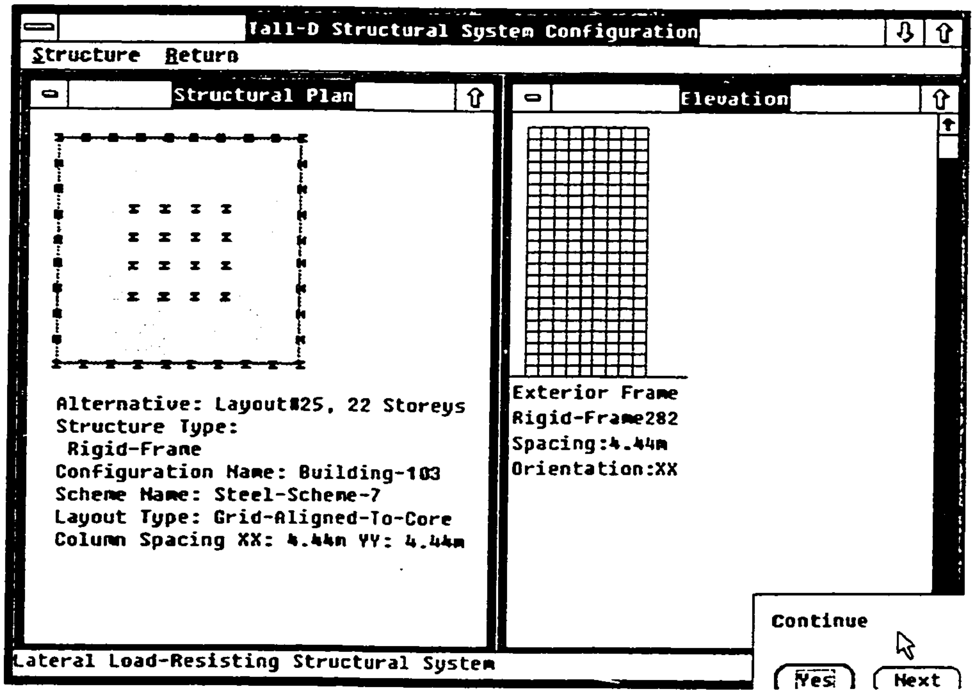


Figure 5.5(k) Rigid frame structural system alternative for Layout#25: Perimeter column spacing 4.44m. Additional core rigid frames.

is removed by the designer. The drawback with this alternative is that the perimeter is not taken advantage of to form a more rigid, economical frame and to remove the rows of intermediate columns between the core and perimeter. The 8m span would be large enough to provide some occupant flexibility.

Two sample structural configurations with rectangular floor plan alternatives are shown in Figures 5.5(o) and 5.5(p). These layouts are presented to indicate the existence of alternatives close to Layout#25 in terms of the number of storeys as well as net rentable floor area.

Similar, alternative schemes and their respective column layouts are finally generated for overall configurations Layout#14 and Layout#1 and presented in Appendix-B. For Layout#14 which is 15 storeys high, different types of steel framing schemes and column layout plans are shown in pages B-13 through B-16. For Layout#1, which is 11 storeys high, steel framing schemes and column layout plans are presented in pages B-17 through B-20. The structural scheme alternatives generated for these two latter configurations are combinations of different column layouts and column spacings for rigid frames structures. No shearwall nor braced frame alternatives are generated for Layout#14 and Layout#1 due to the moderate height of the buildings, which is 59.2m and 40.7m respectively.

The text output corresponding to Layout#25 and all its related structural schemes are summarised in Appendix B (pages B-2 through B-9). The description of column layout is part of the summary for each type of structural scheme generated by Tall-D.

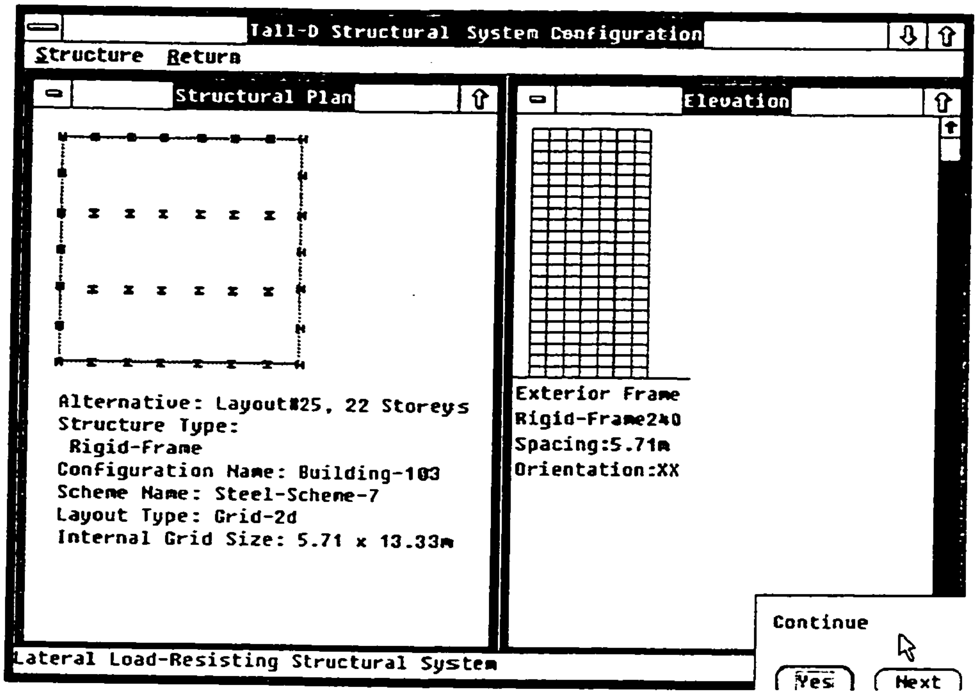


Figure 5.5(l) Rigid frame structural system alternative for Layout#25: Uniform column Grid 5.71 x 13.33m.

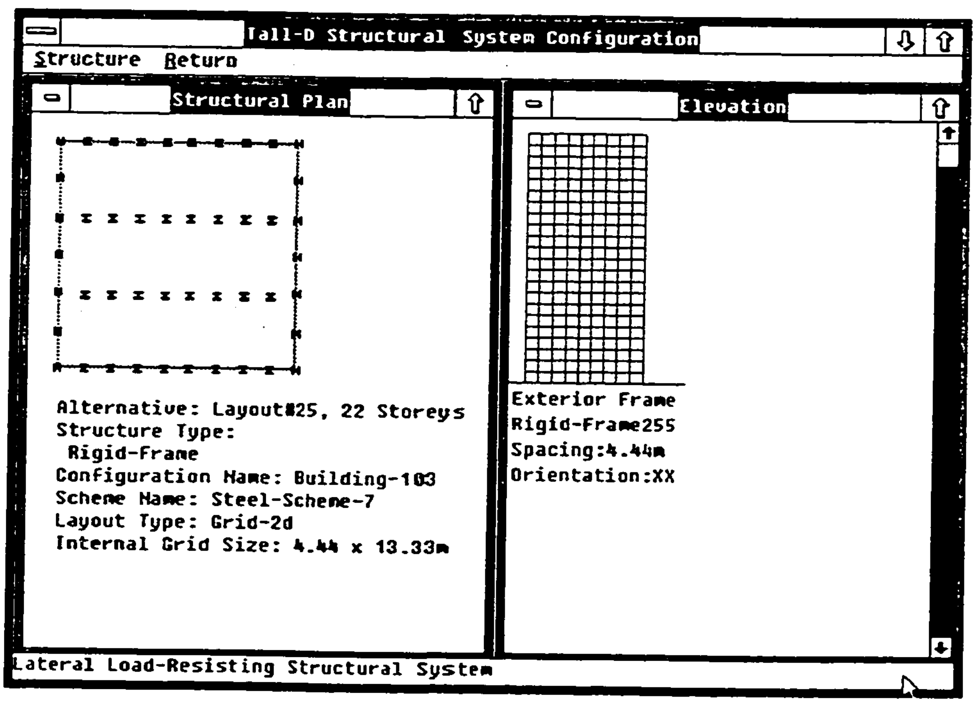


Figure 5.5(m) Rigid frame structural system alternative for Layout#25: Uniform column Grid 4.44 x 13.33m.

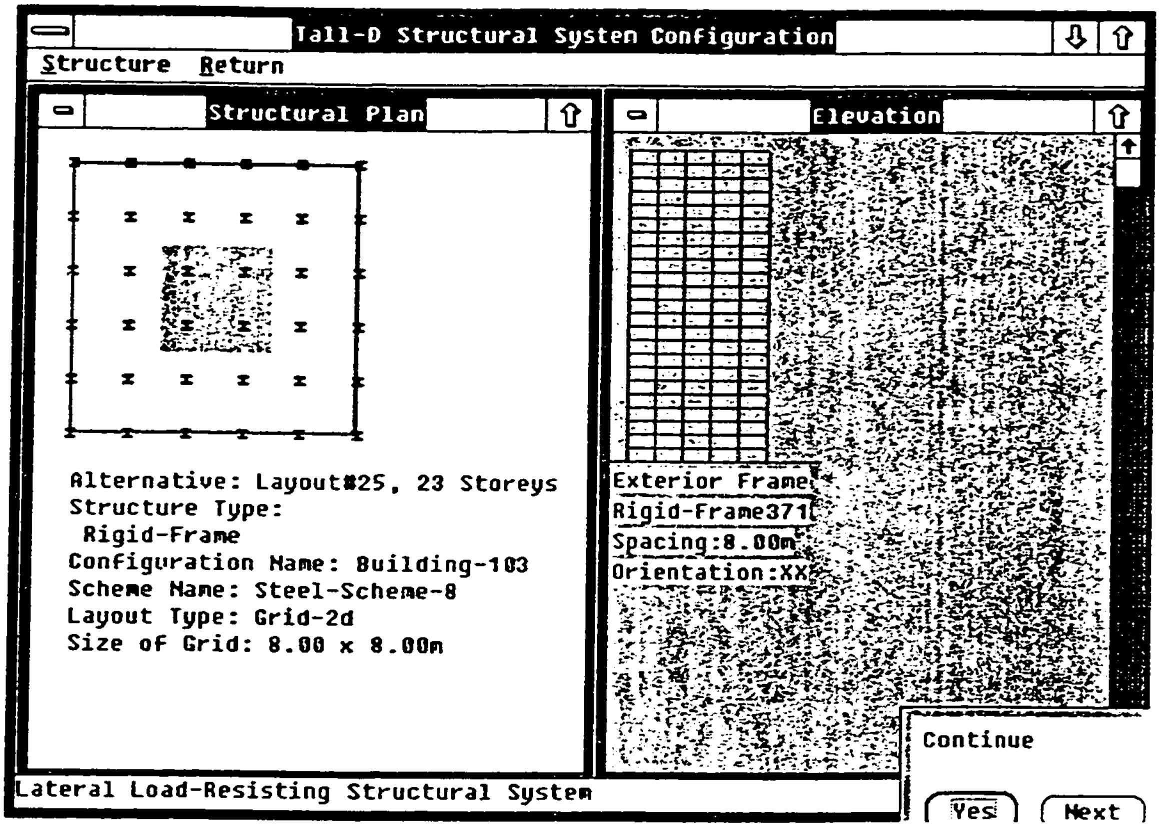
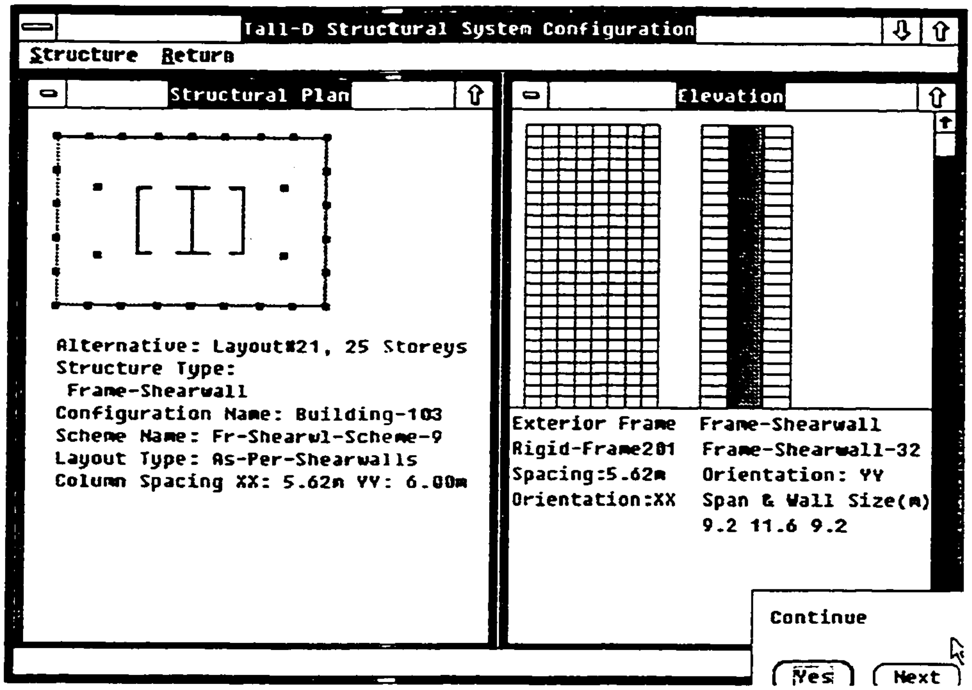
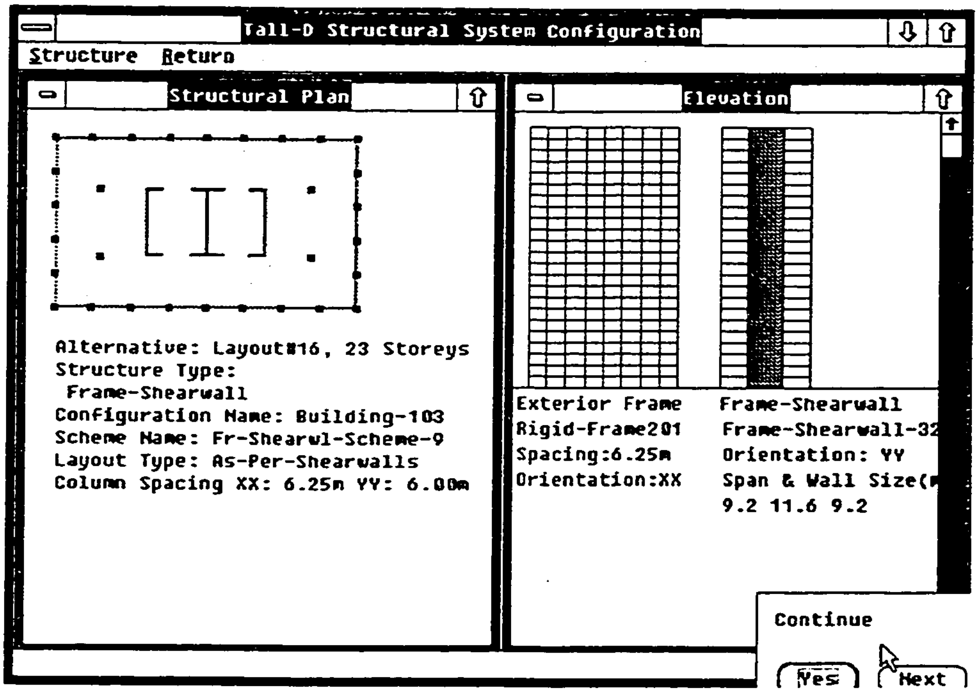


Figure 5.5(n) Rigid frame structural system alternative for Layout#25: Equally spaced column Grid 8 x 8m. Generated when clear span is not mandatory.





Figures 5.5(o) and 5.5(p): Frame shearwall structural system alternatives for Layout#21 and Layout#16 respectively. Both are examples of rectangular floor plans as opposed to the earlier square ones (Layout #25).

5. 2 Gravity Load-Resisting System

The gravity load-resisting system GLS (term used synonymously with floor system) affects the overall design effectiveness of the structural system in a significant manner. The thickness of floor slab, the extent of column-free space, storey height and hence the building cost are some of the factors that are affected by the decisions regarding the selection of GLS.

In the design of GLS, the main parameters are related to the type of floor framing, joist spacing (which is usually related to the planning module adopted for partitioning the office space and installing lighting and other fixtures), ceiling depth and floor thickness according to the material used (steel deck with concrete topping or all reinforced concrete). The overall depth of the floor system is influenced by both the type of vertical structural system as well as the space required for the services that pass above the ceiling. The different types of floor framing considered in Tall-D are presented in the following section along with issues of their integration with building services and LLS.

5. 2. 1 Gravity Load-Resisting System Options

The number and variety of the floor systems available make it an involved task to select the system that best integrates with horizontal services as well as LLS. In Table 5.1, the available concrete and steel floor system alternatives are listed, categorised according to their applicable range of spans. Some of the entries in the table may be applicable for part only of the range of spans listed rather than the complete range.

An example of a design decision regarding the gravity load-resisting system that affects the lateral load-resisting system is presented. In the current implementation, Tall-D begins with a decision about the LLS, then the GLS is selected. However it is also possible to begin with the GLS design. Assuming that a flat-plate (or flat-slab with drop panel) is chosen for its excellent horizontal services integration, then there needs to be either a close enough column spacing or a dedicated lateral load-resisting system such as

Table 5.1 Gravity system options for concrete and steel buildings.

Construction Material	Type of Gravity System	Short Span L ($\leq$ 9m)	Medium Span L (>9m, $\leq$16m)	Large Span L (>16m, $\leq$28m)
Concrete Floor Systems	Flat-Plate	•		
	Flat-Slab	•		
	One-Way Beam-Slab	•	•	•
	Two-Way Beam-Slab	•		
	Joist Slab	•	•	
	Waffle Slab	•	•	
	Band-Beam Slab	•	•	
	Haunch-Beam Slab	•	•	
Prestressed Concrete	Solid Slab	•	•	
	Hollow-Core Slab	•	•	
	Double-Tee	•	•	•
Composite Steel Deck Floor Systems with Steel Beams	Rolled Beam	•	•	
	Light Gauge Joists	•		
	Stub Girder	•	•	
	Tapered Beam	•	•	
	Haunch Beam	•	•	•
	Castellated Beam	•		
	Truss Beam		•	•
	Parallel Beam	•		
Note: • denotes suitability of gravity system option; span L is in metres.				

shearwalls. Thus the selection of flat-slab dictates the characteristics of the LLS, i.e. close column spacing. Large clear spans may not be achievable in all designs. Tall-D can configure the geometry for the gravity structure types listed in Table 5.1 as discussed in section 5.2.3. It can also perform approximate structural member sizing for most of them. Exceptions are given in section 5.3.2.

5. 2. 2 Integration Issues

The gravity load-resisting system is a vital element in multistorey buildings. The service ducts have to be well integrated with the GLS as the ducts often need substantial space between the floor soffit and the ceiling. Without such an integration the floor-to-floor height could become too high to be economical in multistorey buildings. Some of the issues that are considered in developing the knowledge-base of the GLS in Tall-D are presented here. Figure 5.6 shows the GLS with the building ducts and services integrated with the structure to varying degree for each of the different steel floor systems.

a) Integration Issues in Concrete GLS: Integration of services and structure in concrete floor systems can be achieved by selecting appropriate solutions based on the span and the desired (or maximum) level of integration. It is however not feasible to always make a perfect fit to the requirements.

Flat-plate and flat-slab: Flat-plate and flat-slab (with drop panels) floor systems are excellent from the point of view of flexibility of services integration. Due to the absence of beams, services are free to span out in any direction without the need for bends underneath beams and girders. However this system is limited to smaller spans where it forms part of a lateral load-resisting system. This system cannot provide adequate stiffness for buildings greater than 15 to 20 storeys (Taranath 1988). It is also not desirable in high seismic areas. Larger spans up to 10m can be achieved with prestressing. Band-beam slabs, where a shallow wide beam runs between the columns, also can be used in buildings where floor-to-floor height is critical.

Beam-slab: Two-way and one-way beam-slab floor systems are less desirable from the point of view of integration of services. This is due to the fact that ducts have to be taken around the beams, which is not always possible without increasing the storey height. In multistorey buildings, the main ducts may run parallel to the deep beam, with the distribution ducts going below the soffit of main beams thereby increasing the storey height to the least adverse condition. Since beam-slab floor systems are unavoidable in

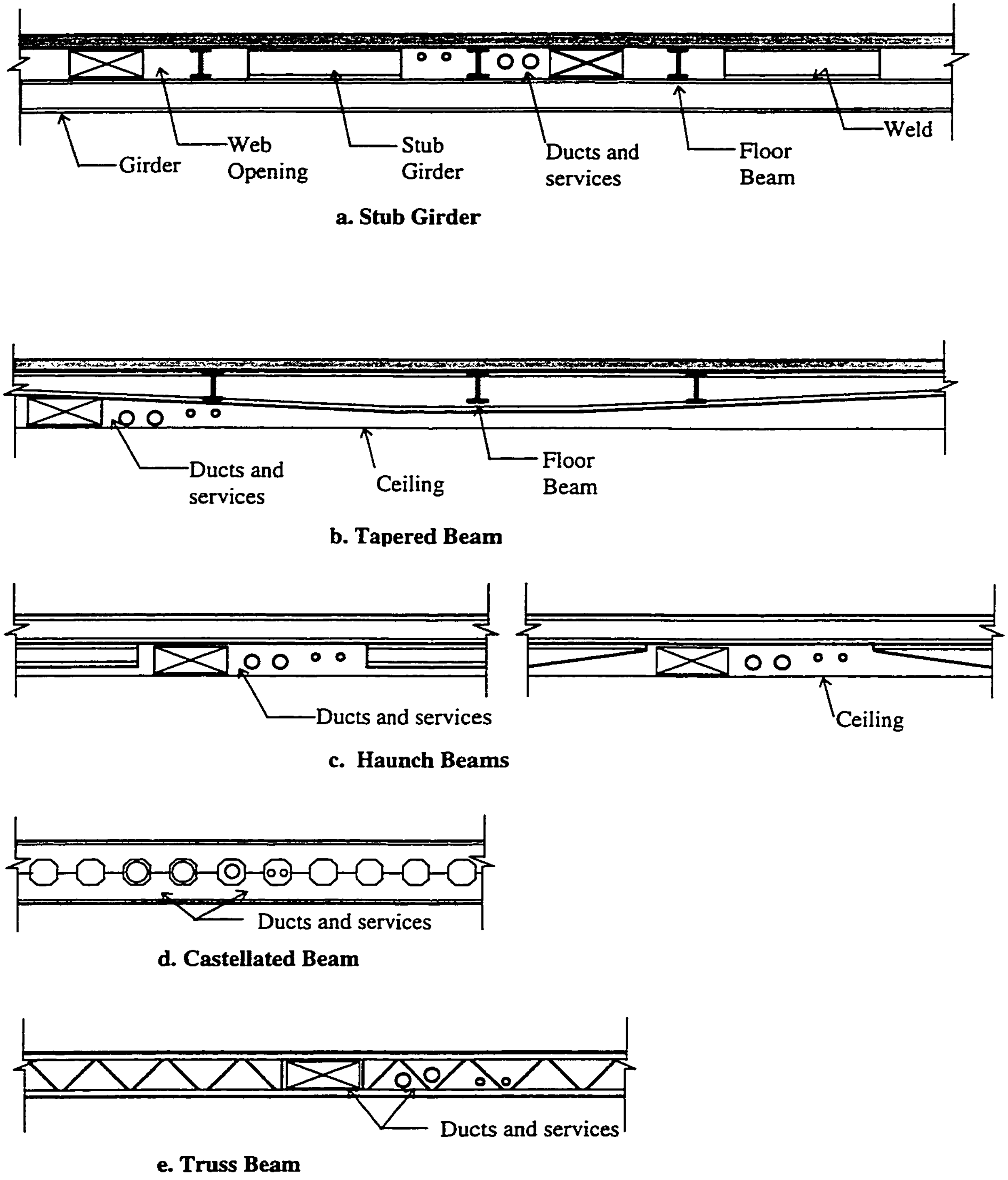


Figure 5.6 Integration of horizontal services with structure (Rush 1986, Trebilcock 1991a)

many cases where rigidity of such frames is required for structural purposes, some depth of the storey height will have to be set aside for the duct space.

Joist slab: One-way joists with the same depth as the beams result in thin floor slabs. The interference to the run of horizontal services is minimal compared to beam-slab systems. In some instances alternate joists are eliminated (skip joists) to make the system more economical. Joist slabs are also good for vertical services distribution where there is a need for openings in the floor (Bennett 1991).

Waffle slab: Waffle slab floor systems are formed with closely spaced joists spanning in both directions. This system requires more expensive formwork, therefore is economical only for longer spans or heavy loads (Bennett 1991, Allen and Iano 1989). The integration potential for this scheme is moderate. One of the advantages is the absence of beams since the floor system spans two ways with columns on a grid. There is a solid shear head around columns and sometimes solid slab bands between columns. Waffle slab is structurally very rigid with little interference to the services distribution. In terms of overall floor depth including service ducts, this system's depth is more than that of a flat-slab but less than that of a beam-slab system. However, on longer spans, waffle slab would be economical since it is lighter in weight than flat-slab and beam-slab systems (less load on the columns), and can span more than flat-slabs.

Haunch-beam slab: Haunch-beam slab systems have deeper sections at the ends framing into the columns, thereby providing higher structural rigidity to lateral loads as well as resistance to high fixed-end moments (Taranath 1988). Since the same beam strength is not required at the mid-span of beams, it is possible to reduce the depth thereby providing a space in the floor system that is available for the duct space. Thus this system often results in an economical solution from the point view of structural performance as well as storey height in cases where the spans are large. Therefore the integration potential of this scheme is high.

b) Integration Issues in Steel GLS: Steel floor systems in multistorey buildings are common. Steel decks are often used as permanent formwork or as a contributing structural part. Steel decks are topped with concrete, normal or light weight according to the design conditions. Steel studs welded along the beams provide the shear connection between steel and concrete, thus resulting in a 'composite' structural action, and an economical floor system. One estimate of the difference in material between non-composite action and composite action floor systems is 30%, with the cost difference being only 15% due to the cost of welded studs (Trebilcock et al., 1991a). The integration potential of steel deck systems with horizontal services is dependent on the beam types that accompany the deck slab system, as discussed below.

Rolled beam: In this system the beams in the floor framing are rolled sections or built-up sections. The integration of horizontal services in such a beam-column steel structure with deck slab is about the same as a concrete beam-slab schemes described in previous section. Minor service ducts can penetrate the beams by way of stiffened openings in the web at appropriate locations, however with a certain loss of structural efficiency. Such an approach may not be feasible if the beams are moment-connected to the columns with a view of making up a lateral load-resisting system. In any case, major ducts cannot penetrate the structural depth of the floor system, thus increasing the storey height when passed below the soffit of the beams and girders. There are however adaptations of this system which are described below that try to overcome the need to reserve part of the storey height exclusively for the horizontal services.

Stub girder (Fig. 5.6.a): One such system that incorporates the duct space within the structural depth of the floor system is the stub girder. It is formed by welding at some intervals approximately 1m-long rolled I-sections (stubs) on top of the girder to act with it providing the shear connection between the girder and concrete slab, to give composite action. The gaps between the stubs are used to support the beams in the other direction as well as to run HVAC and other ducts. These openings are large enough so that the stub girder system provides for excellent integration of horizontal services. Structurally it is most efficient for large spans in the range of 12-15m. However they have been designed for spans for up to 27m (Trebilcock et al. 1991a). The stub girders can also be made stiff enough to account for moment-connections at girder ends, if designed as part of the LLS.

Tapered beam (Fig. 5.6.b): Such beams have a taper of 5 to 6 degrees in the bottom flange, so that the beams become shallow at the supported ends (Trebilcock et al., 1991a). These are suitable for use in simple frame construction (assuming the presence of a lateral load-resisting system). The shallow parts of the beam can be used to accommodate the horizontal service ducts without increasing the storey height.

Haunch beam (Fig. 5.6.c): Steel haunch beams have additional structural depth at the ends that connect to the column. The purpose is to increase the moment capacity of the beam where it is required in a moment-connected frame (Taranath 1988). The reduced depth of the beam at the mid-span provides space for the horizontal services distribution, without the need for additional depth that would add to the storey height. Though it results in increased fabrication costs, haunch beams are economical in larger spans.

Castellated beam (Fig. 5.6.d): These beams are made out of rolled sections cut and rewelded in such a way to form openings that can accommodate horizontal services. There are limitations as to the size of the openings since these would affect the stiffness of the system. For this reason, castellated beams are good only for light servicing. They are inadequate in more than moderate spans or where there are point loads, and thus can only be used under certain conditions (Trebilcock et al., 1991a).

Truss beam (Fig. 5.6.e): One of the frequently used systems in multistoreyed buildings is the versatile steel truss and deck slab composite construction. It is an excellent system from the point view of services integration. It consists of trusses spanning the large span. The steel deck spans the distance between the trusses as secondary beams do. In a truss deck system the large space between trusses can be used to run the main distribution ducts. In the orthogonal direction, since the truss is made of linear elements, there is ample open web space to accommodate the secondary distribution ducts. Though the structural depth of the truss system is larger than many other alternatives, it results in good overall depths due to the accommodation of services completely within the structural depth of the floor system.

5. 2. 3 Geometric Configuration of Gravity Load-Resisting Systems

The total depth of the floor system is arrived at by calculating the slab thickness, beam/girder depth, then adding the space required by the services. It is therefore necessary to estimate the sizes of the structural elements of the floor system and also to define a value for the depth required for the duct space. Two representative cases from among the many possible floor systems are presented here.

Figure 5.7a shows the approximate thickness of a one-way spanning solid slab as related to the span, adapted from charts available for different types of concrete and steel construction (Allen and Iano 1989). Adapted from the same source is Figure 5.7b showing the depth of steel beams and girders. The relations are implemented by means of a straight line equation so the particular function returns the thickness given the span of the slab or beam. The value returned is rounded up to the next 10mm and in concrete to the nearest 50mm. The equation used for the above type of slab is implemented by the Lisp routine `One-Way-Slab-thick (Span)` shown in the Appendix C (page C-48) along with the related routines implementing the procedure for the gravity system object. Similar Lisp routines are incorporated in Tall-D system for the different gravity systems listed in Table 5.1.

An extension of current work could incorporate such refinements or details as the depth of service ducts and more specific geometric parameters such as the location of secondary beams, joist spacing and spans for beams and slabs.

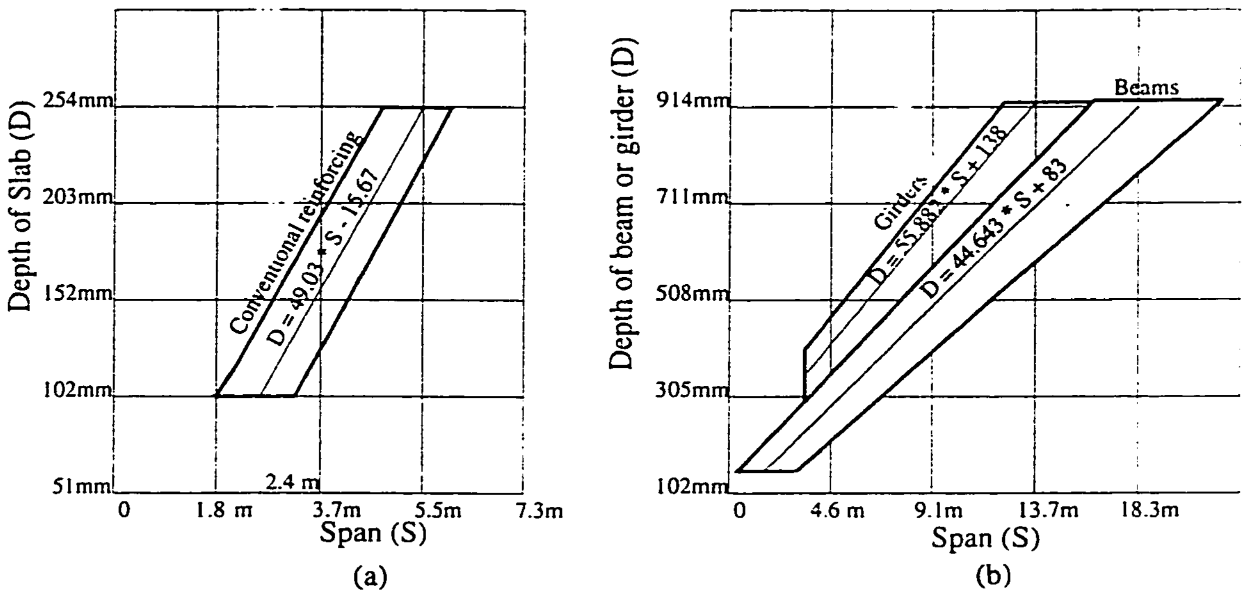


Figure 5.7 (a) Thickness of one-way solid slab. (b) Depth of steel beams and girders (adapted from Allen and Iano 1989).

5. 2. 4 Participation of GLS in Lateral Load-Resistance

In resisting lateral loads, the main function of the GLS is to act as a stiff diaphragm in its own plane and therefore to distribute the lateral loads to the vertical structural elements.

For rigid frames/semi-rigid frames, depending on the required lateral support, selected frames can be moment-connected so as to make them part of the LLS. Some large span gravity systems such as stub girders and haunch beams can participate in lateral load-resistance (Taranath 1988). However as the structure grows taller, a designated LLS becomes necessary. This is because a widely spaced column structure is not stiff enough to provide adequate lateral support.

Flat-plate, flat-slab floor systems may be considered as participating in lateral load-resistance only with small column spacing. In high seismic regions and where the span is larger than 6m, the presence of beams between columns gives desirable lateral support.

The presence of mechanical levels may be exploited to limit lateral drift in zones of high lateral loads, by placing outrigger and hat trusses that mobilize the active participation of perimeter columns in lateral load resistance as discussed above in section 5.1.

5. 3 Preliminary Design of Structural Components

Preliminary design of structural components is an important part of system Tall-D in the generation of building design alternatives. The implementation of some of the design methods is discussed in the following sections. Issues in the implementation of the rest of the methods are discussed here although not completely implemented in Tall-D.

5. 3. 1 Loads on the Structure

Codes specify a minimum level of loading and method of distribution on the structure (NBCC 1995). For example, to estimate gravity loads in tall buildings to design columns, a load reduction from the upper floors can be applied. The assumption is that there is a very low probability that all the floors are fully loaded at any given time. Codes also specify geographical zones in Canada to determine wind and seismic load intensities. Tall-D incorporates these values for the city of Montreal. A complete design implementation would incorporate such code-specified load sets of live loads due to occupancy, snow, wind and earthquakes as well as dead loads due to self weight of structure, finishes and other permanent accessories in the building. The computations involved in the generation and application of loads on the structure do not follow a simple automatic procedure. Dead loads largely depend on the initial selection of structural members. The coefficient values used in expressions for calculating lateral loads due to wind and earthquakes for preliminary design purposes as well as their distribution require that subjective decisions are made regarding stiffness, mass, terrain exposure and so forth. Additional load considerations due to long-term deformations of structural members also require judgement on the part of the designer with respect to connections. Tall-D however is concerned with the rapid generation of alternatives and implemented to consider loading data commensurate with the methods of initial member sizing in Tall-D.

Tall-D estimates the gravity load as well as the lateral wind load according to the provisions of NBCC (1995). While the gravity load calculation along with tributary area provisions for column load reduction in multistorey buildings is implemented in Tall-D, the calculated lateral wind load pressures at two-third height of the building are not fully integrated with the portal method of lateral load analysis. One reason is the type of member sizing procedures in Tall-D which do not require moments for every column and beam in the structure as described in section 5.3.2. Seismic loads are not calculated by Tall-D because initial member sizing is based on gravity loads followed by (or in combination with) a proportionate correction for lateral load effects without regard to the exact moments caused by the seismic load. As shown in Chapter 6, this approach has resulted in member sizing compatible with those in case studies and has proved suitable for a system like Tall-D. This helps in the objective of rapid generation of alternatives that are at the same time accurate enough to form the basis of relative comparison of alternatives as well as to proceed to a more detailed design stage. To appreciate the type of tasks implemented in Tall-D with regard to loading, consult the names of related functions in Tall-D listed on pages C-28 and C-29 of Appendix C.

5. 3. 2 Initial member sizing

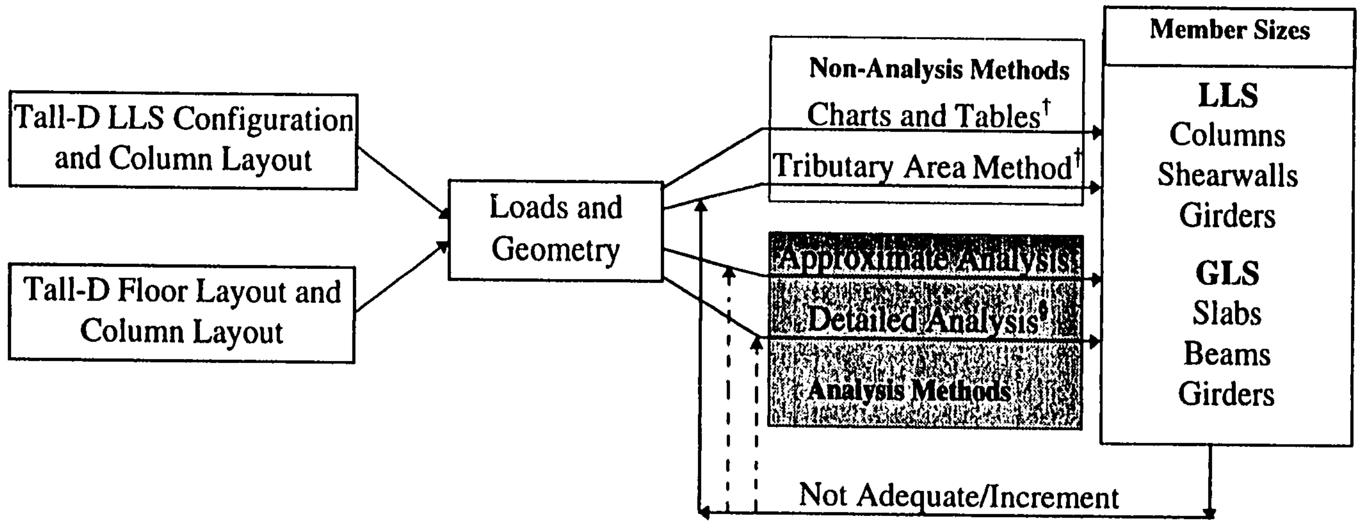
In preliminary design, one needs to establish approximate sizes of structural members - columns, beams and shearwalls - to initiate the first design/analysis cycle. Following a first estimate of loads as specified by Codes, initial member sizing is the next logical step in any structural design process as shown schematically in Fig. 5.8. This step is concerned with the selection of relative stiffness for each member, rather than absolute values. A good initial assumption of the cross-section, i.e. close to final size, will reduce the number of analyze/design cycles in the subsequent sizing process. In this pre-analysis sizing of members, simple charts and rules-of-thumb are used (Allen and Iano 1989, Trebilcock et al. 1991a and 1991b, Schollar 1989, Bennett 1991 and Moreno 1985). For example, to determine GLS component depths, spans for beams and slabs are used. Similarly, to determine the sizes of column and shearwalls, tributary areas of columns and shearwalls are used to quantify vertical loads. Charts, and methods that are based on such indirect measures of loads and stresses in building structures, are of particular value in this initial phase of member design. Fig. 5.8 shows a schematic of the member sizing philosophy in Tall-D as part of the bigger picture. Tall-D implementation of initial member sizing supports the different types of structural components listed in Table 5.2.

Tall-D approximates the size of columns after calculating their tributary area according to NBCC. The cross-section is augmented by 50% to account for lateral-load resistance in the case of rigid-framed buildings and of brace-framed buildings in the planes where braces are located (Nilson and Winter 1986). A correction is applied to the size of the section to account for concrete strength. The chart (see Fig. 5.9) uses a concrete strength default value of 35MPa. Higher strength concrete is used as the number of storeys increase from 15. The correction factor for cross-sectional area of columns is 1.25 for 20MPa concrete, gradually decreasing to 0.75 for 75MPa concrete. In the case of shearwalls, the tributary area calculation is extended to estimate gravity loads. The section at the base is increased iteratively from an initial trial section until the compressive stress is less than the allowed value for the selected concrete strength. The section is then increased in size as an approximation against lateral loads (Coull and Stafford Smith 1991). As an example, the chart used for determining the initial size of concrete columns is shown in Fig. 5.9. The Lisp code that implements this procedure is presented in Appendix on pages C-40 to C-44. Following this in the same appendix, the

Table 5.2 List of structural components Tall-D generates initial sizing for.

GLS components	GLS and LLS common components	LLS components
Flat-Plate	Rolled Beam	Steel columns
Flat-Slab	Light Gauge Joists	Concrete columns
One-Way Beam-Slab	Stub Girder	Concrete Shearwalls
Two-Way Beam-Slab	Tapered Beam	Bracings*
Joist Slab	Haunch Beam	
Waffle Slab	Castellated Beam	
Band-Beam Slab	Truss Beam	
Haunch-Beam Slab	Parallel Beam	
Solid Slab	Double-Tee	
Hollow-Core Slab		

*Bracings put in place, not sized.



†Implemented in Tall-D (see Table 5.2 for different list of structural components).
‡Portal method implemented as a stand-alone module.
§No detailed analysis nor link thereof in Tall-D.

Not part of Tall-D

Figure 5.8 Overview of possible structural member sizing methods for LLS and GLS with Tall-D implemented methods identified.

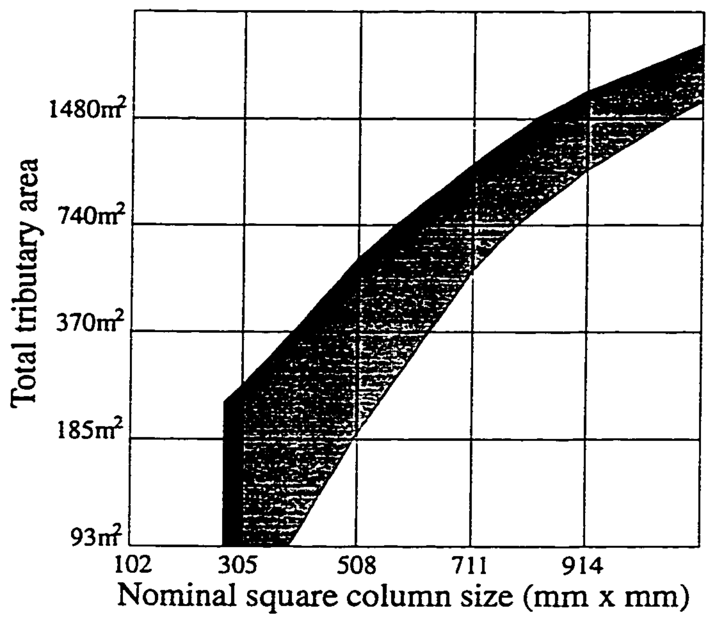


Fig. 5.9 Initial sizing of concrete columns based on tributary area (Allen and Iano 1989).

Lisp code for steel columns initial sizing is presented on pages C-44 to C-47. Tall-D uses similar LISP functions to implement calculation-oriented tasks typical in engineering design applications. For a quick survey of the number and variety of tasks implemented in Tall-D with regard to member sizing, the reader is referred to Appendix C for relevant rules in Tall-D on pages C-3 to C-6 and related functions in Tall-D on pages C-28 through C-31.

For compatible floor systems (slabs, beams and girders), the column layout is searched to find a controlling span. This span is used as the primary factor to arrive at the initial depth of floor beams. The corresponding span for the slabs is used for the concrete slab or steel deck depth. Together the beam and slab depths provide a value for the nominal depth of the floor system. Office occupancy does not have a great variation in loading intensity. In general, this size may be augmented by 50% as a first approximation for preliminary design to account for lateral-load moments in only those frames where the beams are expected to participate in the lateral-load resistance (e.g. rigid frames). However in Tall-D all gravity system components are initially sized only for gravity loads as they are not allowed to participate in lateral-load resistance, except for a few systems like the haunch girder system. In general, the additional depth of floor elements due to lateral load is justified only on the rare occasion after making use of the perimeter beams first. In laterally braced systems such as steel brace-framed buildings and shearwall buildings, no such augmentation of the initial size of the floor system is required due to the fact that beams are not subjected to moments from lateral loading.

For the example building corresponding to Layout#25, the initial sizing information is presented in Appendix B (pages B-5 through B-10) for the different steel and concrete design alternatives shown in Fig. 5.5. In the case of steel structures, nominal size of column is tabulated at different storey levels as well as column locations. For concrete structures, the column size of a square column is similarly tabulated. The shearwall details if present are summarised in a separate section.

The gravity system alternatives are summarised in Appendix-B, Table B1 (see pages B-11 and B-12). At the beginning of the table, the first five concrete gravity system alternatives for the frame-shearwall scheme are presented. As mentioned, the main design factor is the depth of the gravity system that is either determined by the depth of the main beam alone or in combination with the slab depth. The second part of the table in the following page presents the gravity system alternatives for the three different steel LLS. Here again the first five candidate alternatives are presented in each case. The ranking is based on known in-place cost data (Means 1995). Tall-D uses the unit cost to generate relative ranking of GLS alternatives. Since the 'Perimeter-Braced' alternative GLS spans are identical to that of 'Braced-Internal' scheme, the GLS for the former is not presented.

5. 3. 3 Use of Approximate Analysis

Though the use of approximate methods in structural analysis and design apparently decreased after computer-based software was introduced, it has remained the only way of performing a quick check of computer results. Conventional analysis software (such as frame analysis and FEM packages) comes with a predefined set of elements. It is time-consuming to model a structure at the preliminary design stage with these elements and difficult to obtain reasonably accurate results from the model due to many reasons. First the amount of input data describing the discretised structure is generally considerable. Second the intricacies of modelling structural behaviour with general purpose elements often lead to a less than ideal FEM model. Then the sheer number of alternatives being considered at the preliminary stage preclude a conventional analysis. Simplified methods of analysis that are suitable for building structures and based on rigorous theoretical models are one way of addressing the problem. There is renewed interest in approximate methods of analysis and design (Monasa 1991, Coull and Stafford Smith 1991, PCA 1992). Such analytical modelling better serves the specific needs of the different structural systems than the general purpose analysis software in the context of preliminary design. Full-fledged analysis for all the different alternatives at the preliminary design stage is simply not viable.

Knowledge-based systems for structural design applications have been reported where structural analysis was either by means of traditional matrix methods (Kumar and Topping 1988), or absent altogether (Maher and Fenves 1984, Sriram 1986). Approximate methods were not taken advantage of. In Tall-D, though the intent was initially to implement a comprehensive set of different methods, only the portal method of approximate analysis has been implemented (see section 5.3.5). The portal method is suitable for analyzing rigid-frames and can also be used in the approximate solutions for many other LLS as discussed in section 5.3.4. Although the cantilever method could also be used, no further discussion is presented as no approximate analysis methods are used in Tall-D.

5. 3. 4 Approximate Analysis Method in Tall-D

Tall-D implementation currently supports analysis of rigid frames by the portal method. However as already mentioned, Tall-D uses the tributary area method described earlier for sizing of LLS components like columns and shearwalls. Table 5.3 shows some of the available textbook methods of hand calculations for approximate analysis of specific tall building structural systems. These approximate methods are often the only means of checking the results of computerised analysis methods such as finite element or other displacement methods. These methods arise out of simplifying assumptions but provide results that serve preliminary design tasks. In Tall-D, the portal method for rigid frame analysis is implemented. This module in Tall-D can take a set of lateral loads and generate the moments due to that load. As can be seen from the table, the portal method can be used in the approximate analysis of different types of structure. The reason for this is that those methods depend on superposition of the behaviour of two or more simpler systems to simulate the overall system behaviour. One of the components resulting from such decoupling is often a plane frame that could be analyzed using the portal method. Thus the purpose of implementing the portal method in Tall-D was to provide the basis for many other approximate methods of analysis. However to completely implement the approximate methods of analysis would increase the scope of this project unreasonably. An extension to the current work could include the implementation of approximate methods identified in Table 5.3 and help improve the accuracy of beam member sizing as discussed in the Tall-D validation section (6.3.7).

The Lisp code of the portal method implementation is presented in Appendix C (see pages C-51 to C-54) along with the routine that calculates the windward and leeward wind pressures on a building according to NBCC code provisions. Also included on page C-55, Figure C-1 from Tall-D that shows a twenty-five storey building analyzed for a uniformly distributed lateral load with the column moments due to lateral loads. Design moments should also include gravity load moments. Then the columns can be designed for combined axial loads and moments. However Tall-D does not perform column design by these steps that are typical at the detailed design stage. Column and beam sizes

Table 5.3 Approximate methods of analysis for specific tall building structural systems.

Structural System	*Approximate Method*	*Information*
Rigid Frame	Portal method	Coull and Smith (1991), p.141-146 Taranath (1988), p.473-474
Frame-ShearWall Interaction	Rigid link at top and portal method for frame portion.	MacLeod (1990), p.140-143
Braced Frames	Beam analogy of parallel chord trusses.	MacLeod (1990), p.124-127 Eligator et al.(1990) p. 17-20
Tubular Frame	Channel analogy and portal method for frame in web portion.	Taranath (1988), p. 489-496

generated by Tall-D are in most cases close to those predicted by an expert or to the as-built sizes, as discussed in Tall-D validation in section 6.3.

Portal method for rigid frames is inadequate when designing multistorey buildings of more than 25 storeys in general and 35 at the maximum (Taranath 1988), due to the significance of the P-Δ effect as height and slenderness of the building increase. Accurate final sizing of members should consider the strength and serviceability requirements (like storey drift) in a comprehensive manner as specified by the Codes. This implies the use of an elastic analysis with provisions to modify member forces for P-Δ effects. Ideally, as part of the final design, an inelastic analysis may be performed. Conventional analysis programs without inelastic analysis capability may also be utilised for plastic analysis by simulating with a series of dummy elements with negative stiffness (Coull and Stafford Smith 1991). The seismic forces may be applied as equivalent static loads as permitted by the Codes (CSA 1994).

In some structural types, simplified approaches are available to calculate drift and P-Δ effects (Robertson and See 1987). For instance, the Code of practice for concrete (CSA 1984), based on results of elastic analysis, specifies the magnification effect of inelastic behaviour and lateral drift of rigid frames. These simplified methods may be used as alternatives to detailed analysis to estimate the lateral drift and its effect on member forces.

In a situation where the designer wants to use a regular structural analysis package to perform a detailed analysis, the DXF interface (see section 3.5.1) provides a route that can be used to readily transfer the geometry. With further refinements, the interface may be developed to transfer structural element properties through DXF files or other file formats.

5. 3. 5 Member Grouping

It is necessary in practise to define structural member sizes in groups to improve constructibility as well as economy of design. For example, in sizing columns and beams of a reinforced concrete frame on consecutive storeys, identical dimensions are chosen for members in many storeys (two storeys to a maximum of five storeys), so that formwork can be optimised by repeated use. Figure 5.10 shows the column grouping and dimension information for the structural alternative generated by Tall-D for the example discussed in section 5.1.3 for the specific steel design alternative shown in Fig.5.5(m). Since formwork can represent a significant part (up to 15% of structural cost) of the construction costs, it is crucial to rationalise dimensions and detailing in high-rise structural concrete (Anthony 1985).

Tall-D groups members along the height of the building. The concrete sections are incremented in steps of 50mm. Only a defined sub-set of all steel sections is used for steel structural members. The GLS designs by Tall-D are repetitive along the height of the building. These features contribute to constructibility and overall economic construction. Therefore considerations such as constructibility and economy imply that practical/optimal sizes be obtained on the basis of economy of scale, repetition and simple detailing, rather than optimizing individual members. Previous design systems for buildings (Jain et al. 1993, Haber and Karshenas 1990, Jayachandran and Tsapastaris 1988, Maher and Garza 1996, Paek and Adeli 1988, Sause 1989, Sriram 1986) have not come close to specifying structural member sizes for such an elaborate set of LLS and GLS structural types, let alone considerations of constructibility and economy aspects such as member grouping.

Structural Details:Columns

****

Designation of Building Configuration: Building-103

****

Approximate Column Sizes

Structural Scheme: Steel-Scheme-8

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 4.4m Aisles: 13.3m

STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
23-22	(W150 150)	(W150 150)	(W200 200)	(W150 150)
21-19	(W200 200)	(W200 200)	(W250 250)	(W200 200)
18-16	(W250 250)	(W250 250)	(W310 310)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Figure 5.10 Column groups and section sizes for alternative Steel-Scheme-8 with Grid-2D column layout (see Fig. 5.5m for a schematic).

5. 4 Summary

The structural module of Tall-D demonstrates the utility of a computer-based tool for preliminary design of tall buildings. Generation of relevant alternatives enables the designer to investigate a wider range of alternatives at an initial stage of design than is otherwise possible. Such tools as Tall-D will potentially free designers from routine work allowing more time to be spent on innovative and creative work, until such time when computers can indeed help them there as well.

Tall-D generates LLS by beginning with different column layouts based on specified design constraints. Approximate sizes for these vertical elements are arrived at, and relative material quantities for the alternative LLS are presented. The GLS that are compatible with the LLS are generated and ranked according to known in-place cost. Construction integration issues have been discussed with respect to the different LLS and

GLS. An illustrative example with different structural alternatives was presented. Initial member sizing and subsequent refinement by way of analysis using idealised structural models was discussed along with the implemented approximate method of frame analysis - the portal method. Graphical display of alternative column layouts in plan and the corresponding elevations of the building are produced by the Tall-D system.

Thus the structural component of Tall-D presented in this chapter breaks new ground by way of a practical design tool as well as in the underlying formalism and implementation. Two domain areas have been integrated successfully. The knowledge-base in Tall-D, though not all-inclusive, is a contribution to formalising a significant part of the structural design domain. A hybrid system of knowledge representation has been found superior to purely rule-based systems. The physical design entities are represented as objects (also referred to as frames) and the design knowledge is represented as production rules. Objects facilitate the attachment of many procedures specific to each object as well as procedures common to a set of objects. The rules have been grouped into sets, each focusing on a particular aspect of the design process. Integration of KBS techniques, object-based computation and graphical user-interface in a comprehensive manner has been achieved, resulting in a practical design tool to explore building alternatives considering criteria from architectural planning and structural design.

In the next chapter, an evaluation of the Tall-D system by experienced designers is presented to serve as a validation for the system, but also to highlight the usefulness and novelty of the system for preliminary design exploration.

CHAPTER 6

Application and Validation of Tall-D System

Validation of the Tall-D system is presented, primarily based on the collaboration of two external experts. In addition to the design example discussed in Chapters 3 through 5, three new test-cases proposed by the participating experts are introduced as part of the validation process.

6. 1. Validation Methodology

6. 1. 1. Issues in Validation and Verification

Validation and verification (V & V) are two terms that are generally differentiated from one another. Validation is considered a check on software performance based on output, i.e. performance-oriented. In contrast, verification is a check for conformance to system specifications established at the design stage, i.e. implementation-oriented. The former is a functionality check and the latter, a "structural" check.

Validation and verification of knowledge-based systems represents an active research area (Gupta 1993). In the early days of knowledge-based systems, validation and verification was done on an ad hoc basis. Then methods and tools were developed to check for redundant and missing rules in knowledge-bases (Cervera 1993). Since formal V & V is performed against an initial set of specifications, there is a need for formal specifications as well as the definition of acceptability criteria of a KBS. V & V methodologies from conventional software engineering can also be adapted for KBS (Sommerville 1993). Cervera states that V & V should not be done at final development stage alone, but used in combination with structural-validation techniques such as static and dynamic inspection. There is nonetheless a lack of widely accepted KBS life-cycle protocol that establishes which V & V activities can be used and when such activities can be performed. Current research on standardisation in software engineering may be extended to V & V of KBS.

6. 1. 2. Validation Methodology for Tall-D

Although the knowledge-base of Tall-D could still be further refined and expanded, it reached a definitive mass permitting conventional validation. Two methods were used in the validation of the Tall-D system: (i) incremental validation by random output checking, followed by (ii) validation by external experts. Since the development of Tall-D was not initiated with formal specifications, verification against such a set of specifications did not come into consideration. Tall-D was checked in an incremental manner, as a portion of source code or module was developed. It was a cyclic process in which coding, checking and debugging were performed by the developer. Random output checking is claimed to be more efficient for catching failures than static checking such as consistency detection (Zualkernan and Lin 1993). Due to the lack of automated V & V tools for KBS available off-the-shelf, the above two conventional methods were used in the validation of current work.

During development, the addition of a new knowledge module (group of rules and objects) is tested for appropriate activation at the overall level. Each module is then gradually developed by adding rules and attributes of objects. Trial solutions for buildings of various heights influenced by the new module are generated. Since it is possible to redefine key input parameters in the Tall-D system, the effect on the building structure due to variations in design constraints such as availability of materials, seismicity and soil bearing capacity can be investigated.

The validation of design alternatives is facilitated due to the graphical interface of Tall-D that displays building plan and elevation with related information based on the internal knowledge and data representation. Accompanying text screens displayed by

Tall-D provide corresponding detailed information. A text record of the design alternative in an output file (ASCII format) provides a permanent record of salient design information.

6. 2 Incremental Validation

Incremental validation for each of the modules in the knowledge-base included several steps as follows:

- Checking the boundary conditions for the input and the corresponding output values generated by the system.
- Tracing the firing of the rules and the values asserted by the rule consequent.
- Examining the explanation for assertions in the knowledge-base, at random.
- Comparing with documented cases of building design to correlate major design decisions made by the system.

Numerical values were checked with hand computations. During the development stage, the objective was to verify that the development process was on track. Different aspects of one module such as rules, objects, member functions and the consequent results were inspected. The overall performance of the system and its functionality were monitored as soon as a major module was sufficiently developed. The knowledge-base of Tall-D was checked with different combinations of input parameters and design options during typical design sessions. Such a checking was done manually during the development of the Tall-D system. The *explanation* facility was used not only during the Tall-D development/debugging process, but also to verify the derivation of slot values of instances and assertions in the knowledge-base. The system generated *explanation* consists of the rule antecedents and consequents as well as any axioms that contributed to the derivation of a particular assertion in the knowledge-base. *Tracing* of the rules and assertions during the execution of the system was yet another method. *Tracing* of selective rules and functions also enables watching the triggering of a particular rule or

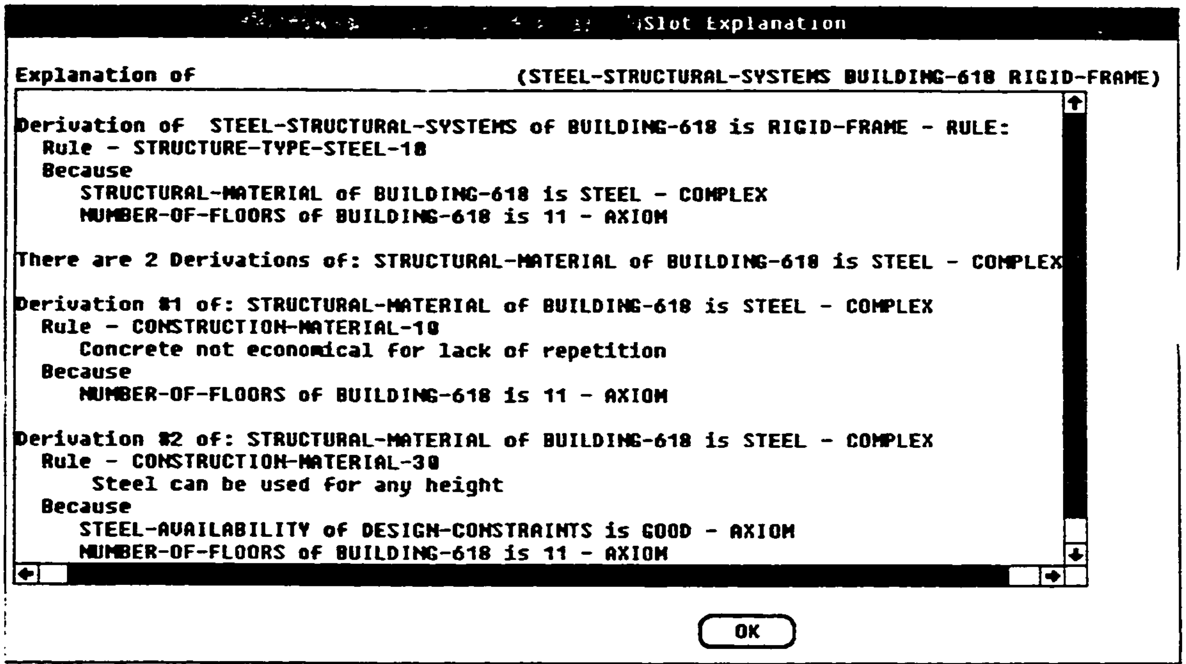


Figure 6.2.1. Use of *explanation* facility to query Tall-D decisions.

the use of a particular function during execution. Vital information regarding a design alternative is available for viewing from Tall-D, some in graphic format and some in text format, by selection through the top menu bar. Some features of the KBS development tool, used during Tall-D development to verify intermediate steps, are illustrated below.

Figure 6.2.1 shows the *explanation* facility being used to query the derivation of the value `Rigid-Frame` in the slot `Steel-Structural-Systems` of an instance of `Building-Configuration` object (`Building-618` here). Such queries were carried out on the different slot value assertions in the knowledge-base to verify their derivation.

Another way that the initial validation was performed was to access complete instances representing the different alternatives to inspect the slot values. Figure 6.2.2 shows a design alternative (instance `Building-535` of the object `Building-Configuration`) with part of the attribute values in the visible window. This focus on complete information with regard to a particular alternative/component enabled the verification of individual design alternatives.

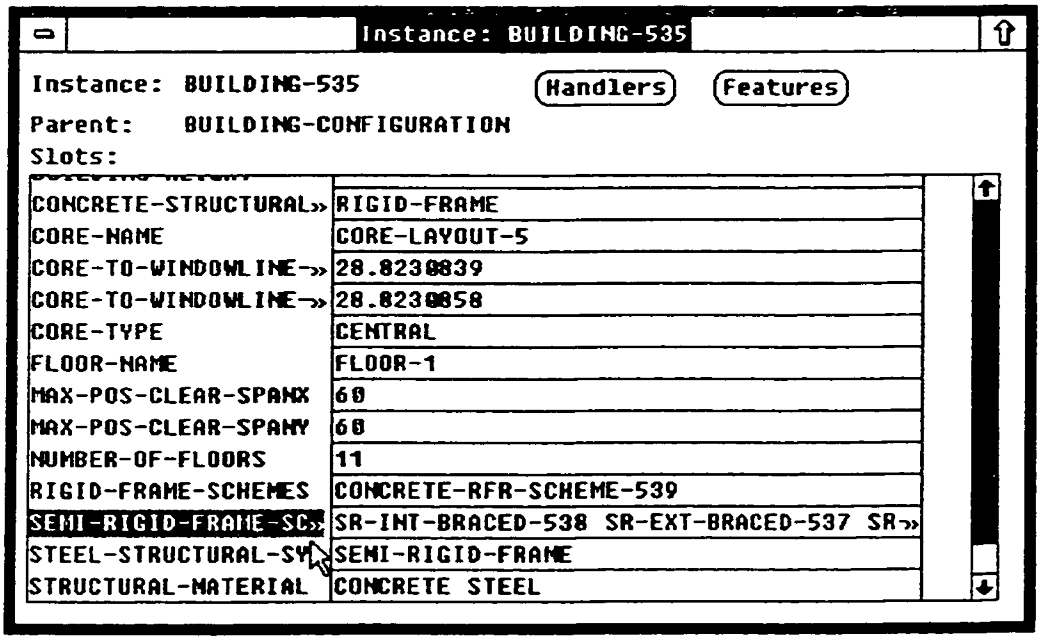


Figure 6.2.2. Direct inspection of instances in a Tall-D design alternative, to check internal values.

The *breakpoint* feature of the development system is another useful feature in the incremental validation of the system. It enables the assertions in the knowledge-base to be inspected at any desired stage of execution, by halting execution of the system. Used along with the *restart* feature, it could be used to branch off into a different design scenario each time.

Figure 6.2.3 shows numeric values printed from lisp routines. They serve to *trace* intermediate values in specific routines by the use of print statements. The system generated *tracing* on the other hand is useful to view assertions either from firing of rules or new values being assigned to slot values of frame instances. For example, the *tracing* of rules can be activated with a command: `(gw-trace :rule)`. Therefore, a trace can help to debug and to observe parts of the knowledge-base used during execution. A specific function can also be traced if necessary. This information is helpful in verifying when and how often a function is invoked.

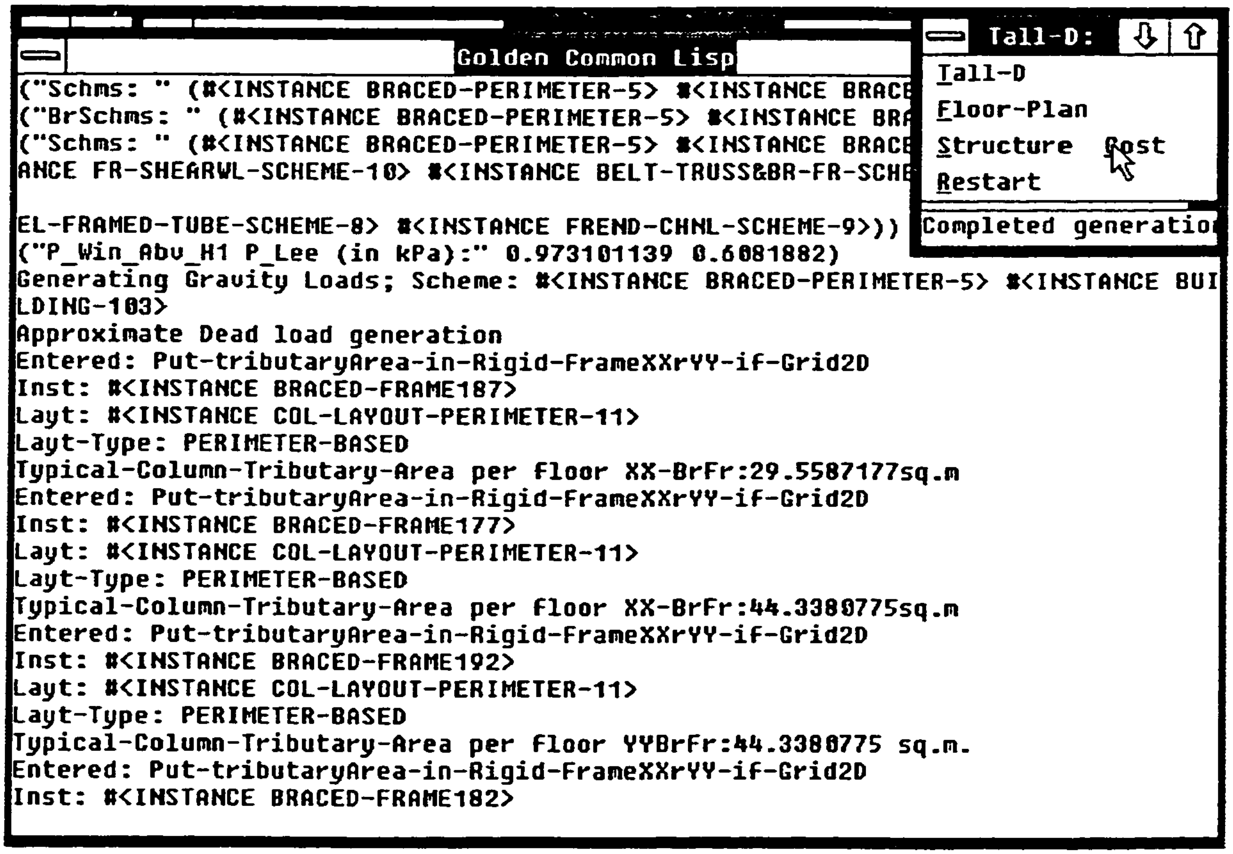


Figure 6.2.3 *Trace* using print statement, showing Tall-D execution of lisp routines to verify intermediate values in numeric calculations.

6. 3. Validation by External Experts

6. 3. 1 Introduction and Objectives

The most important part of V & V was carried out with the participation of two experts in the design of tall buildings. The objective of validation by external experts is to test the Tall-D system with leading designers from the industry who are familiar with the practical aspects of multistorey building design. Also, an overall evaluation of Tall-D concepts and performance from such experts was considered a meaningful measure of the original nature of the work as well as the contribution this work makes in the area of design tools in the less formalised scenario of preliminary design. Therefore an architect and a structural engineer who had worked on some important tall building projects were invited to test and evaluate the Tall-D system.

Test-cases supplied by the two experts were used to compare the designs with those produced by Tall-D. These design sessions enabled the experts to critique the individual cases as well as the overall performance of Tall-D. The overall performance was evaluated by the use of a comprehensive questionnaire including the different features of Tall-D as an automated design tool. The multi-storey test-cases used are actual buildings: Place du Canada, IBM-Marathon and 1000 de La Gauchetière, all located in Montreal.

Sections 6.3.4 through 6.3.6 present the individual test-cases. Most of Tall-D generated output for these design cases is presented in Appendix D, which is divided in three sections for the three design cases respectively with page numbers for each design case starting with 1. The page numbering convention for these sections is "D1-" followed by the page number, referring to Design Case 1 and so on. It may be noted that in the following discussion, the terms design-case and test-case are used interchangeably to refer to the existing building on site. Experts evaluations using a questionnaire and written comments on the Tall-D system are presented in section 6.3.7.

6. 3. 2 Selection of External Experts

The two experts that volunteered their time and knowledge to evaluate Tall-D were Mr. Antony Niro and Mr. Serge Vézina. Both have extensive experience in the design of multistorey buildings, many of which being high profile projects.

Mr. Niro is an architect consultant (Di Miele Niro Architects, Montreal). He represented Marathon Realty on the IBM-Marathon office tower project in downtown Montreal. He has been an architect on many projects while at Marathon Realty and now practices his profession as part of his own firm.

Mr.Vézina is a structural engineer at Lalonde, Valois, Lamarre, Valois (1993) inc., Montreal, a subsidiary of SNC-Lavalin. He was the principal structural engineer on the 1000 de La Gauchetière project as well as the New Forum project, both recent projects in Montreal. The 1000 de La Gauchetière building is a 47-storey office tower. The Forum is a multipurpose indoor stadium. He was suggested as an external validation expert by the thesis supervisor and after an initial meeting gladly accepted to volunteer time for Tall-D evaluation.

6. 3. 3 Experts Choice of Buildings

The choice of test-cases was left to the external experts with the request that at least one concrete and one steel test-case each be provided. Though it was expected that each will provide the minimum two, Mr. Vézina could only spare time for one test-case that followed a generic demonstration of Tall-D features. He selected his most recent tall building project, the 1000 de La Gauchetière. Though it represented the high end of the height limitations for Tall-D (50 storeys), it was a successful interactive session that resulted in alternative designs from Tall-D as well as it presented the features of the design system to the expert.

In the other session with Mr. Niro, Place du Canada and IBM-Marathon office towers in Montreal were the test-cases. They were selected by the expert perhaps due to his personal knowledge and involvement in the projects, as was the case with the other expert. One common feature of test-cases is that they are all office towers, mostly prismatic with rectangular or square floor plans. Since Tall-D cannot design buildings with other than rectangular floor plans, the test-cases thus satisfied the requirement of the system. The three test-cases are presented in the same order design sessions were conducted with the expert consultants.

6. 3. 4 Design Case 1: Place du Canada

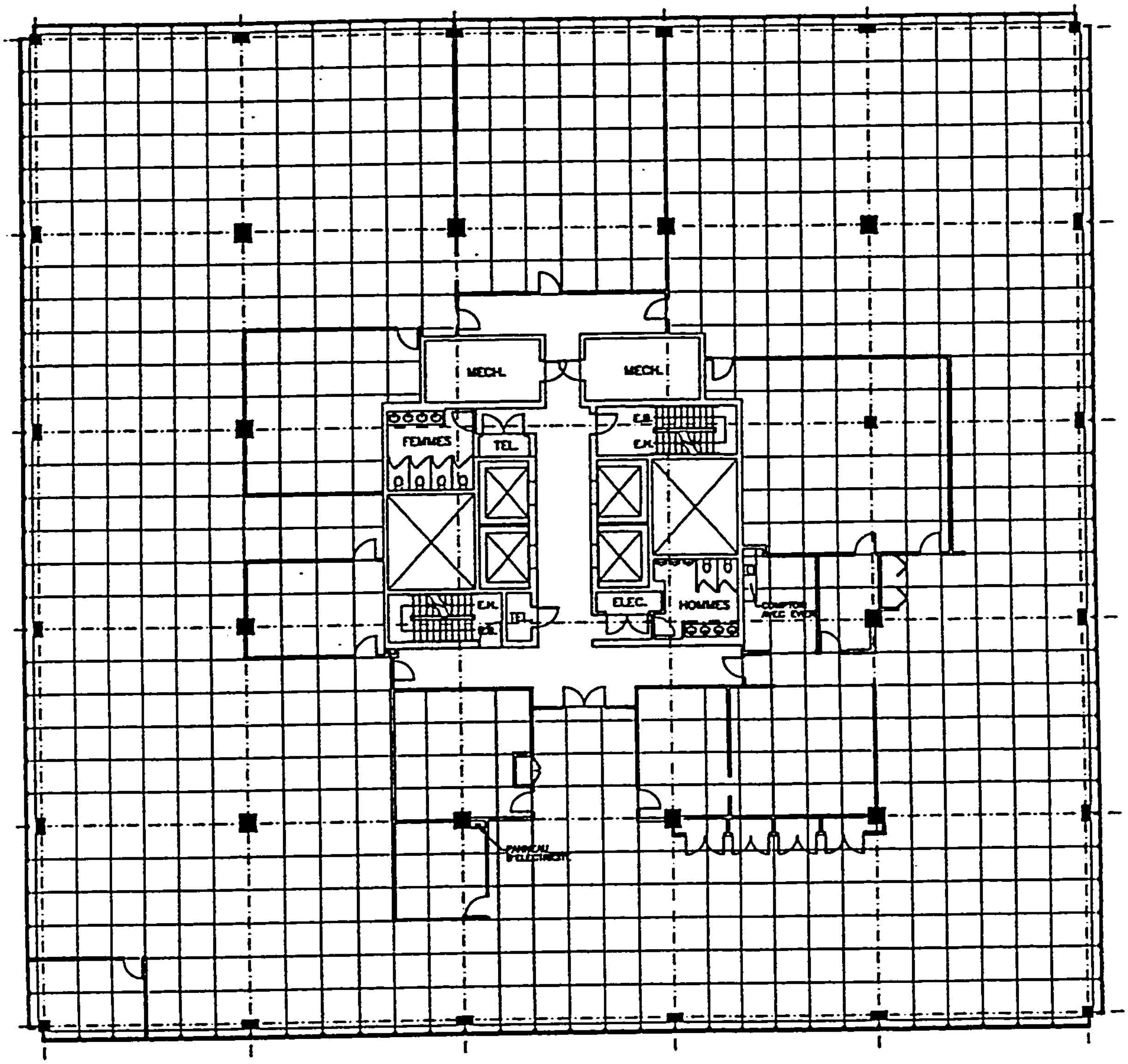
The first design case is a 24 storey concrete office building. Figure 6.3.1 shows the overall plan dimensions of the actual building. Figure 6.3.2 shows the elevation of the same. Table 6.1 shows the input parameters specified to initiate the design case using Tall-D system. The text output file and some graphical screens for this test-case are presented in Appendix D-1. Eleven floor layout alternatives (Layout#1 to Layout#11) remained at the end of Level-I of the initial generate-test process (see Part C, p. D1-4 as well as Fig.D1-2, p.D1-16). These floor layout alternatives are evaluated according to a set of criteria specified by the designer (see Part E, page D1-5). The resulting ranking of the alternatives shows a list of ten layouts with evaluation values close to each other. Except the fifth alternative in the list, all the remaining five have a score between 2150 and 1950. They are all plans with a central core and are consequently ranked close to each other. The first ranked alternative is 14 storeys tall with a score of 2150. Of those ranked second, the height of the buildings ranges from 12 to 23 storeys.

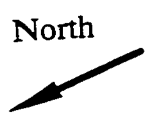
The approximate cost of the building was estimated at 36 million dollars. Fig. D1-2(a) (p.D1-16) shows an approximate cost right at the very beginning of the session when only the required rentable area is known and hence shows a value of 37 million dollars. Fig. D1-2(b) (page D1-16) shows the order of magnitude estimate after core size and the number of storeys was calculated (for the first ranked alternative). The cost according to the designer's estimate is in the range of 42 million. No specific reason for the difference in cost is identified. Cost estimates at the preliminary design stage are generally accurate to about 25% and final cost is often heavily influenced by specifics of each project.

Due to the reason that the number of storeys of the first ranked alternative does not match closely that of the test-case (14 versus actual 24), the expert has picked Layout#8 ranked second (score 2050), with 19 storeys and plan dimensions (55mx40m), to proceed further to structural configuration. The number of storeys in the test-case is 24 including two mechanical levels. The structural configuration of Layout#9 is also investigated by

Table 6.1 Input parameters for Design Case 1: Place du Canada building.

Parameter	Value
Net Floor Area required	33405 m^2
Maximum number of floors	25
Dimensions of Plot	70 x 60 m
Approximate Budget [1995$]	$42 millions





Place du Canada
1010, de La Gauchetière W., Montréal
Year Built: 1967
Ground Floor Plan
Plan Dimensions:43.9x39.6m
Sketch not to Scale.
Source: Excerpt from architectural sketch drawings (A. Niro, Architect)

Figure 6.3.1 Typical plan of Design Case 1: Place du Canada, Montreal.

the designer due to the fact it has 21 storeys. He rejected it in the first place due to the plan dimensions (an oblong 55mx35m) and preferred Layout#8. Layout#8 is the closest in plan dimensions and height to the actual building.

Since the actual building has a row of intermediate columns, 'no clear span' has been specified as a preference, to simulate the same in Tall-D (see Fig. 6.3.1). The system generates an intermediate row of columns only if a clear span is judged as uneconomical. Details of the different structural schemes generated for initial Layout#8 are presented in Appendix D-1 starting with a page of contents. One layout generated by the system is presented in Figure 6.3.3 as a representative sample. The information regarding the layouts is presented in the form of text windows that can be browsed through. The user may also request the structural schemes to be displayed graphically. The text displays are in fact written to an output file that is presented in Appendix D-1 (see for example page D1-8). Though all the structural schemes can be displayed graphically by Tall-D only a couple of systems each for concrete and steel are presented in Appendix D1 for Layout#8. Two perimeter-based concrete rigid frame structural alternatives are presented in Fig. D1-3. A set of steel braced frame structural alternatives are presented Fig. D1-4 (see page D1-17). In all five different LLS schemes are generated for Layout#8 (ignoring 'Braced-Perimeter-5' scheme for which Tall-D did not generate adequate details). See page D1-1 for a list of the alternatives and their page numbers. The respective details regarding structural layout geometry (p. D1-5), column sizes (p. D1-8) and gravity system details (p. D1-14) are presented under the respective layout alternative instance names. For purposes of comparison with the actual building the values for scheme 'Concrete-Rfr-Scheme-10' and the alternative with perimeter column spacing of 4.58m and 5.71m are used. See Fig. 6.3.3 and pages D1-6 and D1-10 for details of the Tall-D generated information used in Table 6.2.

Similar details for Layout#9 are also presented in Appendix-D1. The number of storeys being close to that of Layout#8 (19 versus 21) the type of alternatives generated are also similar with an additional structural scheme of frame-shearwall system being generated. Details of these schemes can be found in appendix pages D1-19 through D1-

Figure 6.3.2 Elevation of Design Case 1: Place du Canada, Montreal.

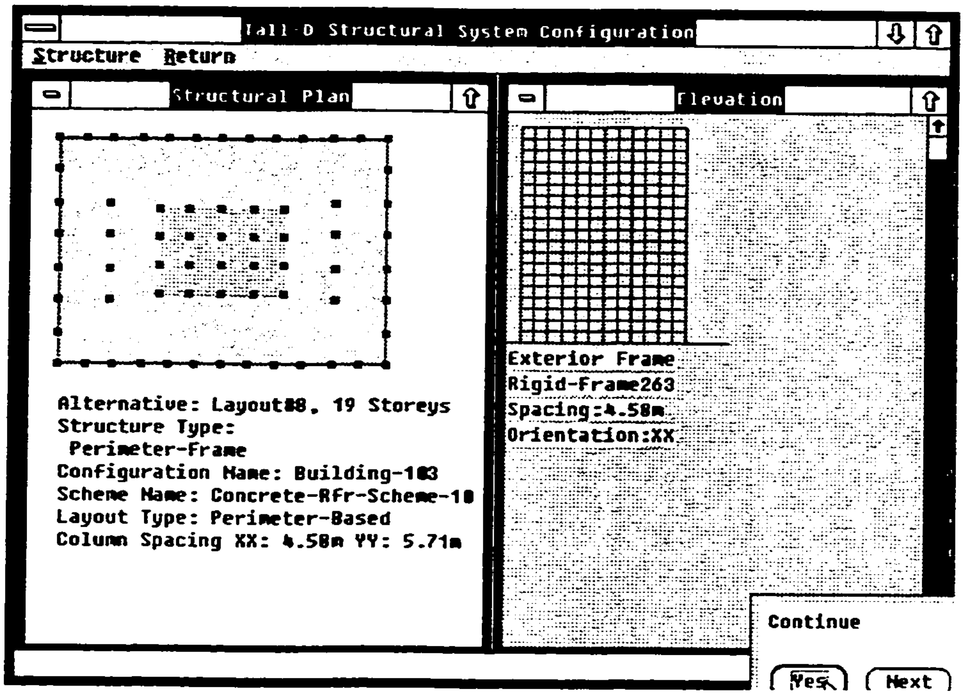


Figure 6.3.3 Rigid-frame structural scheme for Design Case 1, Layout#8.

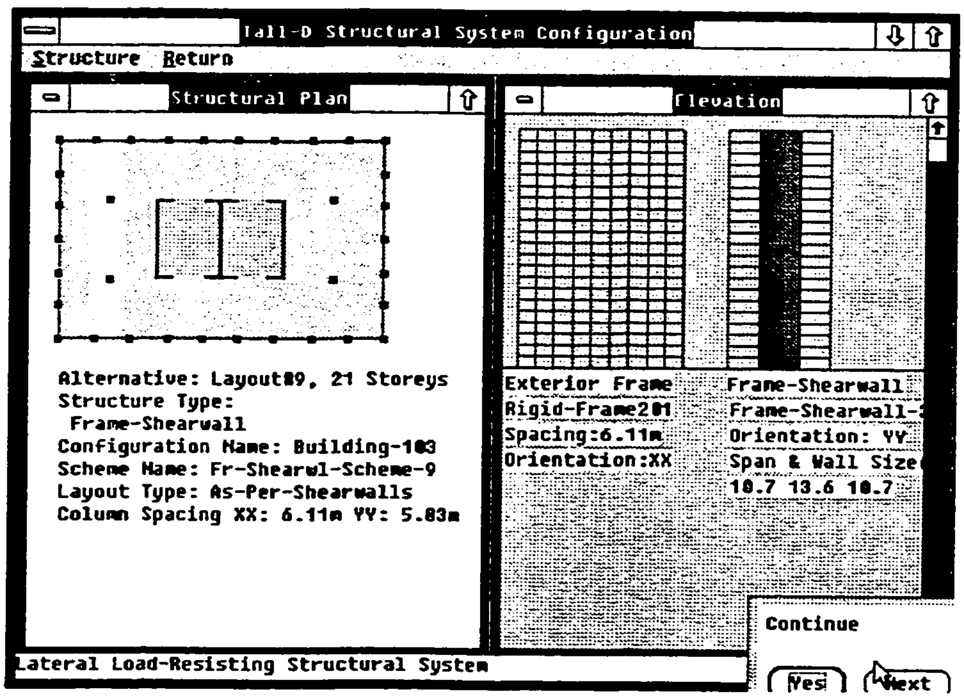


Figure 6.3.4 Rigid-frame structural scheme for Design Case 1, Layout#9.

25. Selected graphical representations of these are also presented in pages D1-26 to D1-28 with a representative layout being presented in Figure 6.3.4. The values for comparison with the actual building used in Table 6.2 for Layout#9 are that of scheme 'Fr-Shearwl-Scheme-9', with perimeter column spacing of xx:6.1m and yy: 5.8m. The Tall-D schematic for this alternative is presented in Fig. 6.3.4. and the shearwall and column details for the same in pages D1-21 and D1-22.

The gravity system alternatives for Layout#8 are presented in Table D1 in Appendix-D1 (see page D1-14). For the different lateral-load resisting structural schemes (schemes imply how the vertical system is laid out in plan), Tall-D generates compatible gravity systems. The table shows the depth of the beam and thickness of slab or deck where applicable. In general the depth of the beam is an indicator as to the overall depth of the gravity system which usually accommodates service ducts. Service ducts are arranged such that they run within the depth of the beams. Where there is no depth of beam mentioned in the table, the system does not need one (e.g. Waffle slab) and the depth of service ducts have to be included to arrive at an overall depth of the system. The table generally lists the top five gravity system alternatives for each scheme. For example for the schemes 'Concrete-Rfr-Scheme-9' and 'Concrete-Rfr-Scheme-10' concrete joist slab is ranked 1 and hollow core slab is ranked 5. The ranking is based on historical data for in-place cost of the floor system but not on the actual quantities for the floor system. The gravity system sizes for the two systems are identical as the internal spans of the gravity system are also identical. The actual building had hollow core slab though the depth of the slab is not known. Tall-D included this alternative but is one of the least preferred one. This is possible due to the fact that the current implementation is based on in-place cost. With its flexibility for service ducts and potential reduction in the floor to floor height of each storey, hollow core slab may prove an economical alternative.

Table 6.2 shows a comparison between the as-built design parameters on the left and the corresponding Tall-D generated parameters on the right. Most of the column layouts (both Layout#8 and Layout#9) are considered reasonable by the expert in terms of occupant space and layout of vertical structural elements in plan. He considered the core

Table 6.2 Comparison of as-built Place du Canada building with Tall-D design.

Building Parameter		**As Built***		**Tall-D (Layout#8)**		**Tall-D (Layout#9)**	
Plan dimensions		43.9m x 39.6m		55m x 40m		55m x 35m	
Core dimensions		25.9m x 8.5m		21.3m x 15.5m		21.3m x 13.6m	
Storeys (+mechanical levels)		22 (+ 2)		18 (+ 1)		20 (+ 1)	
Floor area Gross::Net (m^2)		41722::33405		41800::33696		40425::32706	
Cost of building ($ millions)		42†		37 ‡		37 ‡	
Lateral-load resisting system		Rigid Frame-Shearwall		Rigid Frame		Rigid Frame-Shearwall	
Material grade of columns		n/a		40MPa		40MPa	
Column spacing on perimeter		*Front:xx*	*Side:yy*	*Front:xx*	*Side:yy*	*Front:xx*	*Side:yy*
		8.78m	7.92m	4.58m	5.71m	6.11m	5.83m
Column Sizes on building perimeter (mxm)	Ground Level	0.7x1.0	0.7x1.0	.55 x .55	.55 x .55	.55 x .55	.55 x .55
	Levels 4-6	n/a	n/a	.40 x .40	.40 x .40	.55 x .55	.55 x .55
	Levels 10-12	n/a	n/a	.30 x .30	.30 x .30	.40 x .40	.40 x .40
	Levels 13-15	n/a	n/a	.30 x .30	.30 x .30	.30 x .30	.30 x .30
	Levels 16-19	n/a	n/a	.30 x .30	.30 x .30	.30 x .30	.30 x .30
Number of shearwalls		4		NIL		3	
Shearwall concrete, size and thickness		n/a		-		40MPa; 10.7m long; 375mm thick	
Gravity-load resisting system		Hollow-Core slab		Hollow-Core slab§		Hollow-Core slab	
Clear span (front/back and sides)		7.92m⇕	8.78m⇔	12.3m⇕	8.4m⇔	10.7m⇕	8.4m⇔
Depth of gravity system beams		n/a		420mm **		420mm**	

Notes:

n/a: Information not available.

* Since no detailed drawings were available, the architect approximated the dimensions from available sketches.

† Estimated by architect as the cost in 1995 $.

‡ The number 37 is for the first ranked alternative(Layout# 6) which may have slightly different gross floor area this layout; However it could be used as a common figure for all layouts due to the approximation in the method used.

§ Hollow-core-slab was one among several gravity system alternatives.

** This is depth of hollow core slab and does not include depth of service ducts etc.

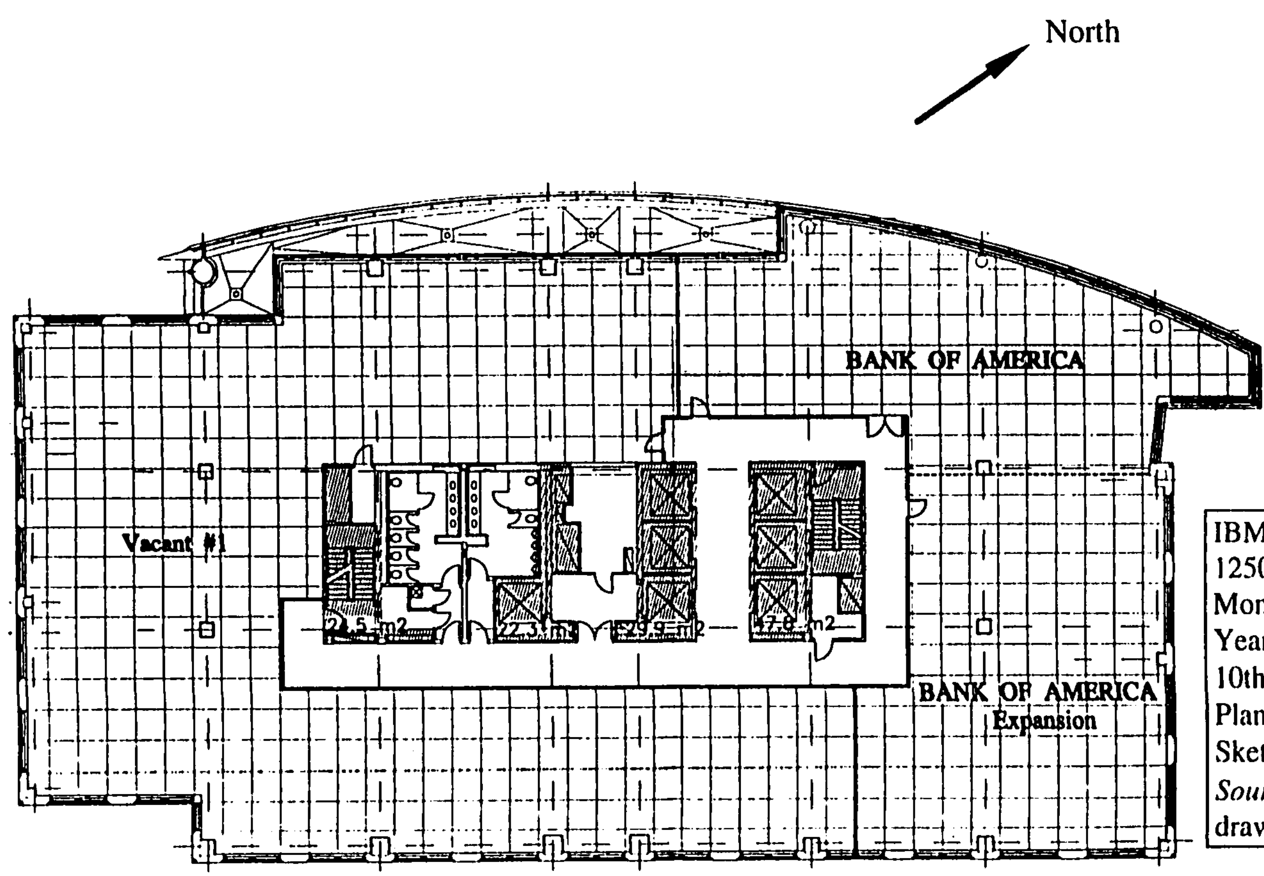
size generated by Tall-D for smaller buildings like this test case somewhat larger than required. The sizes of structural members as well as the spacing of columns are comparable between the test-case and as designed by the Tall-D system. Tall-D has selected a concrete compressive strength of 40MPa. The columns are spaced at between 7.9m and 8.4m in the actual building compared to somewhat closer spacing under Layout#8 and Layout#9 in Table 6.2. The section size of Tall-D columns at ground level are smaller partly due to the closer column spacing and partly due to the higher strength concrete selected by Tall-D. The concrete strength of the test case was not known to the architect. Sizes are measured off an architectural drawing at the ground level, thus explaining the absence of values in the table for column sizes along the height of the building. The clear spans suggested by Tall-D are larger than the test case values, apparently providing greater flexibility of space usage in the building. The size of shear walls in the test case was not known. The Tall-D shearwalls in Layout#9 are 375mm thick made of 40MPa concrete. Overall this test case has demonstrated more to the architect the floor plan and column layout generation capabilities of Tall-D than verified the structural sizing of members due to lack of complete structural plans for this older building (built in 1967).

6. 3. 5 Design Case 2: IBM-Marathon

This building has forty seven storeys including four mechanical levels. The building has a main tower and two low rise annexes attached to the main tower. This test-case is studied considering only the main tower. The base dimensions of the main tower are 60mx30m. The core to window line distance (column free) is 13.7m. The core dimensions are 30mx9m. Figure 6.3.5 shows a typical floor plan of the building under consideration.

Table 6.3 Input parameters for Design Case 2: IBM-Marathon building.

Parameter	Value
Net Floor Area required	65790 m^2
Maximum number of floors	50
Dimensions of Plot	130 x 70 m
Approximate Budget [1995$]	$200 millions



IBM-Marathon
1250 boulevard René-Lévesque
Montréal
Year Built: 1992
10th Floor Plan
Plan Dimensions: 60mx30m
Sketch not to Scale.
Source: Excerpt from space planning drawings (A. Niro, Architect)

Figure 6.3.5 Typical floor plan of building in Design Case 2: IBM-Marathon building.

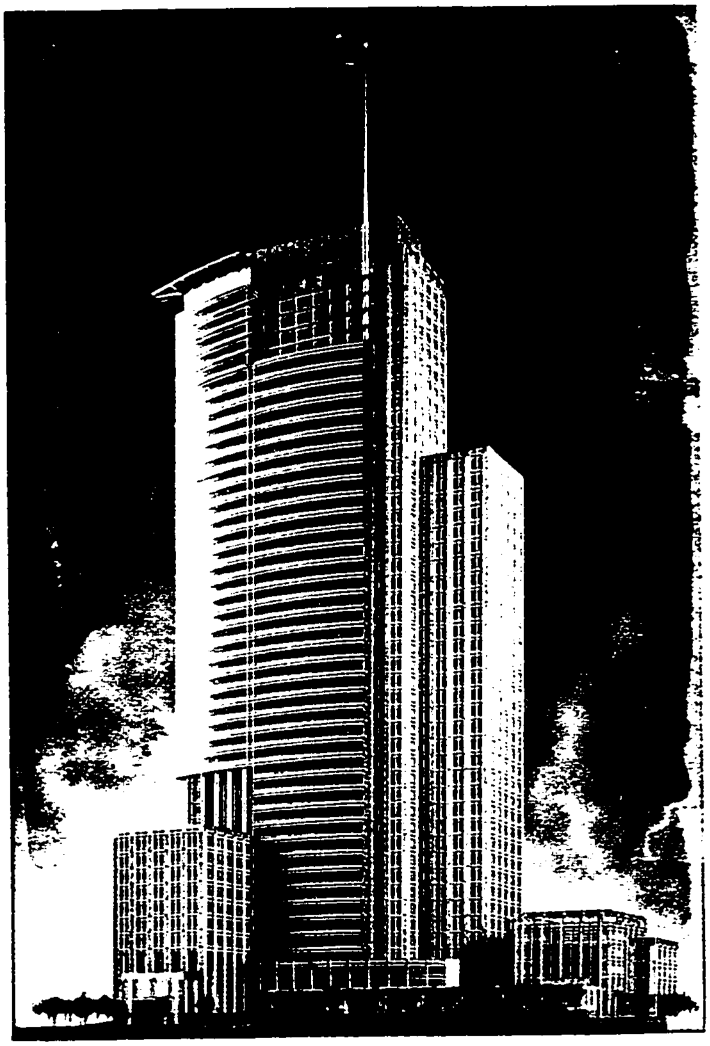


Figure 6.3.6 Elevation of Design Case 2: IBM-Marathon building, Montreal (BOMA 1992).

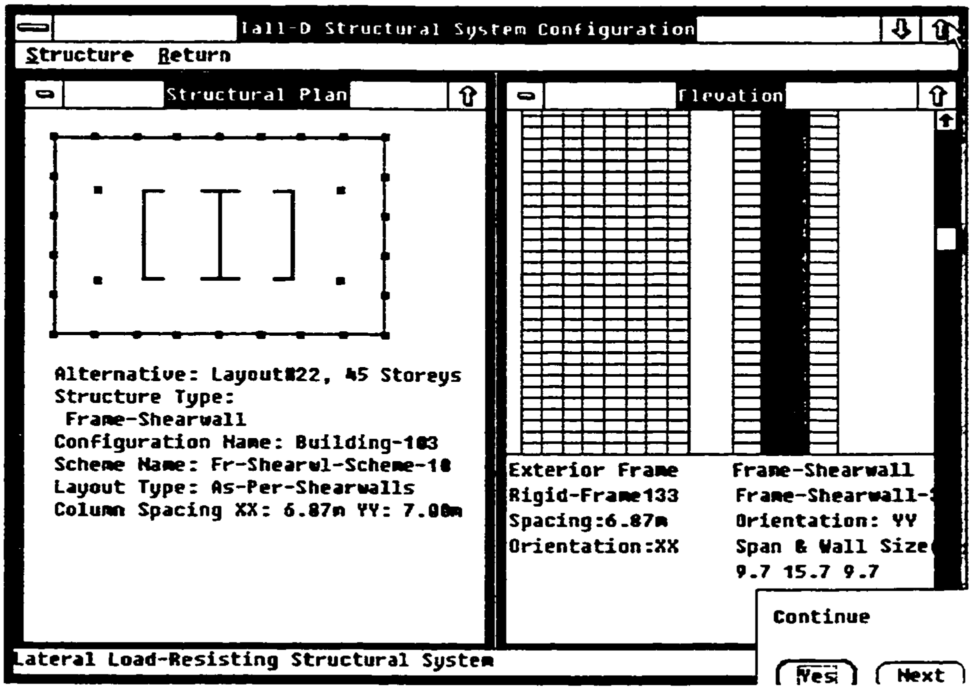


Figure 6.3.7 Structural configuration alternative for Layout#22, Design Case 2.

Figure 6.3.6 shows the elevation of IBM-Marathon building. The designer input to generate possible solutions by Tall-D is shown in Table 6.3. Of the twenty nine alternatives initially generated, twenty three were retained for ranking in Part-D, page D2-4. Complete text echo of the test run for this case is presented in Appendix D-2. Among the evaluated alternatives, Layout#24 ranked second in Part-F, page D2-5, and due to the number of storeys being forty eight and close to the actual forty seven, is selected by the expert to proceed and inspect the geometry. The different structural alternatives generated are included in Appendix D-2 in pages D2-6 and D2-7.

After looking at the column layout and since the plan dimensions of Layout#24 (45m x 40 m) are not close to that of the test-case (which is 60m x 30 m), another alternative Layout#22 ranked third with rectangular floor plan dimensions (55m x 35m) closer to test-case than those of Layout#24, is selected to proceed with geometric configuration and structural design. Figure 6.3.7 shows a structural configuration for Layout#22. The different lateral system alternatives generated by Tall-D for Layout#22 are presented in Appendix D-2. See pages D2-13 through D2-15 for the graphical representation of these alternatives. The descriptive and numeric text output is presented in the same appendix (p. D2-6 through D2-11). In all four alternative structural system schemes are generated, each geometry summarised in pages D2-6 and D2-7. Each scheme has alternative column spacing on the perimeter as noted in the summary. The four types of structural schemes generated are in no particular order, except that the latter two cases were not close to the actual one and not inspected by the expert.

a) All-concrete, frame shearwall scheme. A system that combines shearwalls and rigid frames to provide adequate lateral-load resistance for the taller buildings. The actual building uses this system. Details presented in page D2-6, box-1.

b) Steel framed tube with concrete shearwall. Closely spaced columns on the perimeter along with a shearcore provides great flexibility to design for lateral loads for a tall building like this one. The summary of this system is presented in page D2-6, box-2.

c) All-steel, braced perimeter frames on the exterior faces of the building. Provides greater flexibility for planning occupant space as opposed to the next scheme of interior braced frame. The summary of this system is presented in page D2-7, box-1.

d) All-steel, braced frames on the interior. The bracings are on the inside where they are easier to conceal. However if large open spaces or greater flexibility is needed, this may not be a suitable choice. The summary of this system is presented in page D2-7, box-2.

The details of the column sizes and quantities for the above first two schemes for each column spacing alternative are presented under 'Fr-Shrearwl-Scheme-10' (p. D2-7, D2-8) and 'Steel-Framed-Tube-Scheme-8' (p. D2-9). The other two alternatives being all-steel structural schemes were not inspected by the expert. Two other structural systems are identified by Tall-D as suitable for use in these buildings, but did not complete the design due to the incomplete state of their implementation. These two are framed end channel scheme and belt truss with braced frame scheme.

In a fresh run, Tall-D produced an additional alternative (labelled here Layout#23 with plan dimensions 55mx30m and 49 storeys) which is yet closer to the test-case dimensions, when the percentage of core area (with respect to gross floor area) is reduced to 15% from the previous 20% as suggested by the expert designer. Figure 6.3.8 shows the structural configuration of this layout.

Table 6.4 shows a comparison between the as-built design parameters on the left and the Tall-D generated parameters on the right. For purposes of comparison with the actual building the values for scheme 'Fr-Shrearwl-Scheme-10' for the alternative with perimeter column spacing of xx:6.9m and yy:7.0m are used. See Fig. 6.3.7 as well as pages D2-6 and D2-8 for details of the Tall-D generated information used in Table 6.4. The sizes of the structural members as well as the spacing of columns are comparable between the test-case and as designed by the Tall-D system. Tall-D has selected a concrete compressive strength of 60MPa for Layout#24. The columns are spaced at 7.5m with square sizes of 850mm at base and 450mm at 35th floor. Since the expert preferred the rectangular proportions of Layout#22 the rest of the design was carried out for the same. The corresponding Tall-D values for Layout#22 are 55MPa, 6.9m column spacing, square column sizes of 850mm at base and 350mm at the 35th floor. The corresponding actual values are 55MPa concrete, 9m column spacing, sections of 1000mm at base and 600mm at 35th floor (according to the external expert). The somewhat larger sections of columns for actual design can be attributed to the smaller spacing suggested by Tall-D. The smaller clear span of 9.7m in the design simulation as opposed to the actual value of 13.7m would also impose lesser gravity loads on the perimeter columns, though the occupant space is also less. It can therefore be said that Tall-D columns appear reasonably sized.

Looking through the different boxes corresponding to each alternative structural system and the quantities at the bottom of each such group in Appendix D-2 (p.7-10), one can find the range of quantities used. Though these material quantities are for the LLS alone, they are an indication of the footprint of the structure. It can be said with some degree of confidence that the relative measure of these quantities are an indication of the material costs. Though material costs are not always the binding factor in the selection of a system, all things being equal this factor can certainly be used to select an alternative

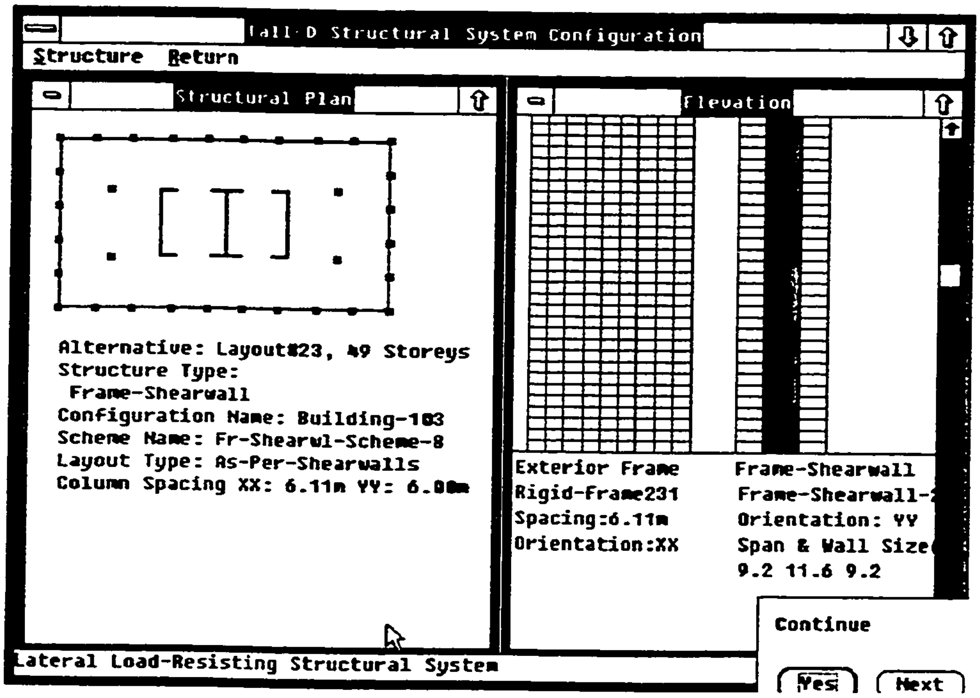


Fig. 6.3.8 Structural configuration alternative for additional Layout#23, Design Case 2.

from those presented by Tall-D. As an example, the alternative with perimeter column spacing of 4.6m and 4.4m on page D2-8 has 143 m^3 less concrete volume than two other alternatives on the same page. Similarly for the steel framed tube alternative the steel tonnage for the columns alone ranges from a low of 261 tonnes to 504 tonnes which makes the alternative with the perimeter column spacing of 7.9m and 8.7m the preferred alternative (see pages D2-9 and D2-10). The quantities are better used as a relative measure than an absolute conclusive parameter. Thus Tall-D generated material quantities provide an evaluation of the structural systems generated, and is appropriate for preliminary design purposes.

The gravity system alternatives are presented in Table-D2, Appendix D-2 (see page D2-11). Concrete joist slab system with a total depth of 1100mm and joist depth 400mm is one of the alternatives for the frame shearwall structural system. It is also ranked first among the alternatives. The other alternatives are, in order, waffle slab, one-way slab with beams, band-beam with slab and finally hollow core slab. The ranking is based on known in-place cost of construction (Means 1996), not on estimated quantities nor on other

Table 6.4 Comparison of as-built IBM-Marathon building with Tall-D design.

Building Parameter		As Built(Tower)		Tall-D (Layout#22)	
Plan dimensions		60m x 30m		55m x 35m	
Core dimensions		30m x 9m 16-35floors	41m x 9m 1-15 floors	24.6m x 15.7m	
Storeys (+mechanical levels)		43 (+4)		43 (+ 2)	
Floor area Gross::Net (m^2)		84600 :: 65790		86625 :: 66168	
Cost of building ($ millions)		(148*)		72	
Lateral-load resisting system		Frame-Shearwall		Frame-Shearwall†	
Material grade of columns		55MPa		55MPa	
Column spacing on perimeter		*Front:xx*	*Side:yy*	*Front:xx*	*Side:yy*
		9.0m	7.5m	6.87m	7.00m
Column sizes on building perimeter (mxm)	Ground level	1.0 x 1.0	1.0 x 1.0	.85 x .85	.85 x .85
	Levels 15-11	n/a	n/a	.75 x .75	.75 x .75
	Levels 25-21	n/a	n/a	.65 x .65	.65 x .65
	Level 35	0.6 x 0.6	0.6 x 0.6	.35 x .35	.35 x .35
	Levels 48-46	n/a	n/a	.30 x .30	.30 x .30
Number of shearwalls		6 at Base; 4 at 43rd Floor		3 throughout	
Shearwall concrete, size and thickness		60MPa; 9m long; 525mm		55MPa; 15.7m; 600mm	
Gravity-load resisting system		n/a		Concrete Joist slab and others (see p. D2-11)	
Clear span (front/back and sides)		13.7m⇕ and 15.0m⇔		9.7m⇕ and 15.2m⇔	
Depth of gravity system beams		n/a		1.10m for joist beam-slab system	

Notes:

* This cost includes the tower, low-rise part, under-ground parking, site preparation etc.; Tall-D cost is only that of the tower, above ground. The actual cost for the as-built tower is 90 million.

† See Appendix D-2 for other alternative systems proposed by Tall-D.

merits of the gravity systems. Similar rankings for the steel framed tube schemes are, again, in order of preference, truss beam, haunch beam, tapered beam, parallel beam and stub girder. The truss beam is 680mm deep with additional steel deck 70mm thick as well an equal depth of concrete topping.

6. 3. 6 Design Case 3: 1000 de La Gauchetière

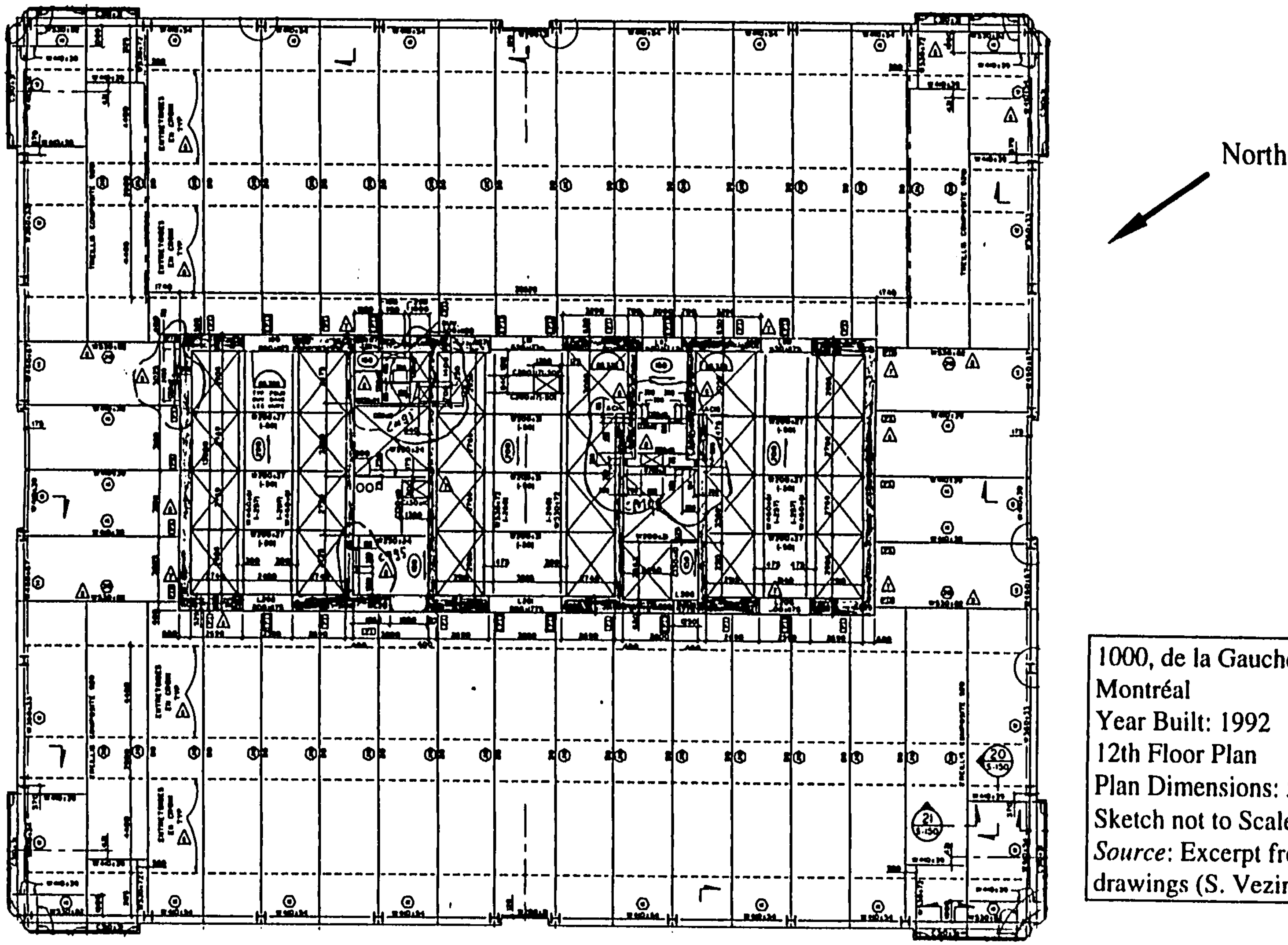
This is another forty-seven-storey building in Montreal completed in the early nineties. The building consists of a plaza and a tower. A typical floor plan is shown in Figure 6.3.9. Elevation of the building is also shown in Figure 6.3.10. The data input to the Tall-D system is shown in Table 6.5. The required floor area specified includes only the area in the tower portion of the building (i.e. excluding the plaza area). Table 6.6 presents a comparison between the actual and a selected Tall-D design. It can be noted that the initial design parameters - entries along the rows for plan dimension, cost of building, floor area, clear span in yy-direction are some parameters that agree with some allowance. Other parameters such as core dimensions, clear span on the sides and number of mechanical levels do not agree all that well, indicating some variation in design approach or scope for improved knowledge-base.

Of the twenty-four alternative floor outlines generated, fifteen are retained after the initial evaluation stage as shown in Appendix D-3 (see Parts B to F, p. D3-3 to D3-6). Ranking of the alternatives is performed as per the preferences of the expert. The ranking of the alternatives is such that two alternatives are ranked first due to equal score. Similarly multiple floor layouts are ranked second and third. Of the two layouts ranked

Table 6.5 Input parameters for Design Case 3: 1000 de La Gauchetière building.

Parameter	Value
Net Floor Area required	100000 m^2
Maximum number of floors	60
Dimensions of Plot	105 x 84 m
Approximate Budget [1995$]	$150 millions

Figure 6.3.9 Typical floor plan of Design Case 3.



1000, de la Gauchetière W.
Montréal
Year Built: 1992
12th Floor Plan
Plan Dimensions: 51mx42m
Sketch not to Scale.
Source: Excerpt from project design drawings (S. Vezina, ing.)

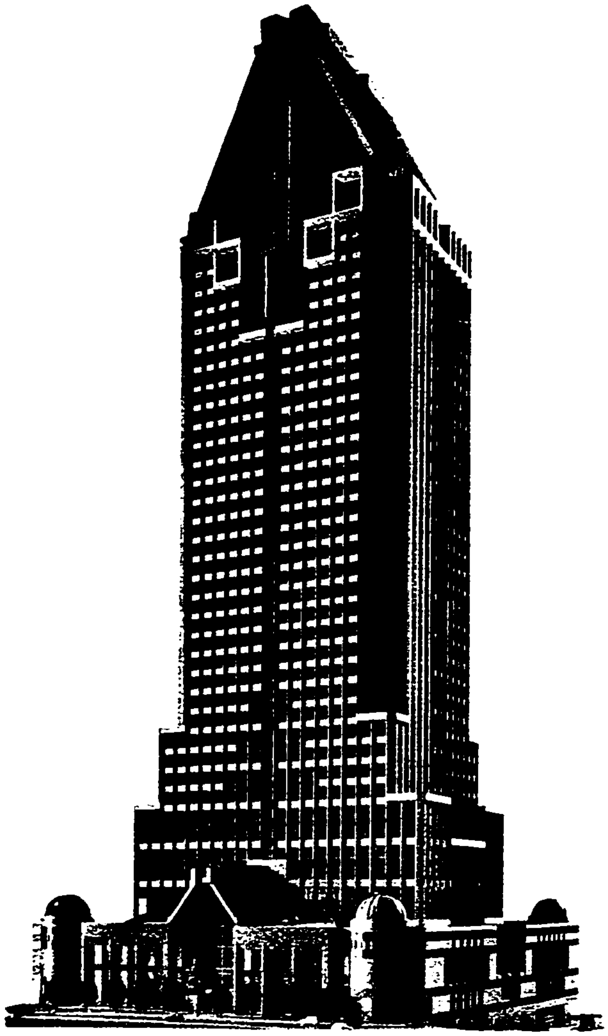


Figure 6.3.10 Elevation of Design Case 3: 1000 de La Gauchetière (BOMA 1992).

first, Layout#14 has forty-one storeys, which compares fairly to the actual forty-seven storeys. However, another plan layout, Layout#15 which is ranked second, is chosen to proceed to the structural design as it is the closest in plan dimensions (55m x 50m versus 51m x 42m actual at the 12th floor level) and number of storeys (45 versus 47 actual) to the test-case.

The alternatives for the structural system consisted of tubular structure with shearwalls in the core of the building. Both steel and concrete columns on the perimeter were considered. The footprint of the steel perimeter columns is however smaller than that of the concrete columns. The test-case has been built with concrete shearwalls in core and steel columns on the perimeter. The footprint of the columns generated by Tall-D system matches closely with that in the site, the actual sections being slightly heavier compared to the Tall-D solution. The complete set of alternatives is presented in Appendix D-3. In all four alternative generic schemes for structural systems are generated, each scheme tried with alternative column spacing. The schemes are:

a) All-concrete, frame shearwall scheme. A system that combines shearwalls and rigid frames to provide adequate lateral-load resistance for the taller buildings. Details presented in page D3-7, box-1.
b) Steel framed tube with concrete shearwall. Closely spaced columns on the perimeter along with a shearcore provides great flexibility to design for lateral loads for a tall building like this one. The actual building uses this scheme, with a slight variation: The perimeter frame has rigid beam-column connections only along the middle third of the building height. Therefore the participation of the perimeter frame in the lateral-load resistance is less than a fully interacting frame-shearwall system, but possibly cost-effective. The summary of this system is presented in page D3-7, box-2.
c) All-steel, braced perimeter frames on the exterior faces of the building. Provides greater flexibility for planning occupant space as opposed to the next scheme of interior braced frame. The summary of this system is presented in page D3-8, box-1.

d) All-steel, braced frames on the interior. The bracings are on the inside where they are easier to conceal. However if large open spaces or greater flexibility is needed, this may not be a suitable choice. The summary of this system is presented in page D3-8, box-2.

The details of the column sizes and quantities for the above schemes (except item c.) for each column spacing alternative are presented under 'Fr-Shrearwl-Scheme-10', 'Steel-Framed-Tube-Scheme-8' and 'Braced-Internal-6' on page D3-9, D3-11 and D3-13 respectively. The braced perimeter option was not selected during column size generation and could not be presented here. Two other structural systems are identified by Tall-D as suitable for use in these buildings, but did not complete the design due to the incomplete state of their implementation. These two are: framed end channel scheme and belt truss with braced frame scheme. Representative Tall-D generated schematics of the alternative structural schemes are presented in pages D3-15 to D3-18.

Figure 6.3.11 shows one of the tubular structural alternatives generated for the building. This particular configuration and column spacing is closer to the actual than any other generated by Tall-D and is therefore used to compare the actual with Tall-D design. Table 6.6 presents a comprehensive snapshot of the comparison for more than a dozen parameters. The following discussion refers to the contents of the table as well as information from the referred to sections of Appendix D-3.

The shearwall thickness is marginally more (675mm for the outermost two walls and 600mm for the single inner wall) than that of the actual (600mm for two outermost walls and 400 mm for all interior walls). The number of walls suggested by Tall-D is less, i.e. 3 compared to 6 actual. However, since the length of the walls in the Tall-D design is more than that of the test-case walls, there is parity in the shearwall cross-sectional area (36.4 m^2 versus 35.4 m^2 actual neglecting the flanges in both cases). Also, the moment of inertia (1140 m^4 versus 472m^4 actual again neglecting the flanges) of Tall-D generated wall sections is more than twice that of the actual. The reason the test-case shearwalls are not as long as Tall-D generated values is due to the difference in the proportions of the core area. The actual test-case core area is thinner (more elongated) than the shape of

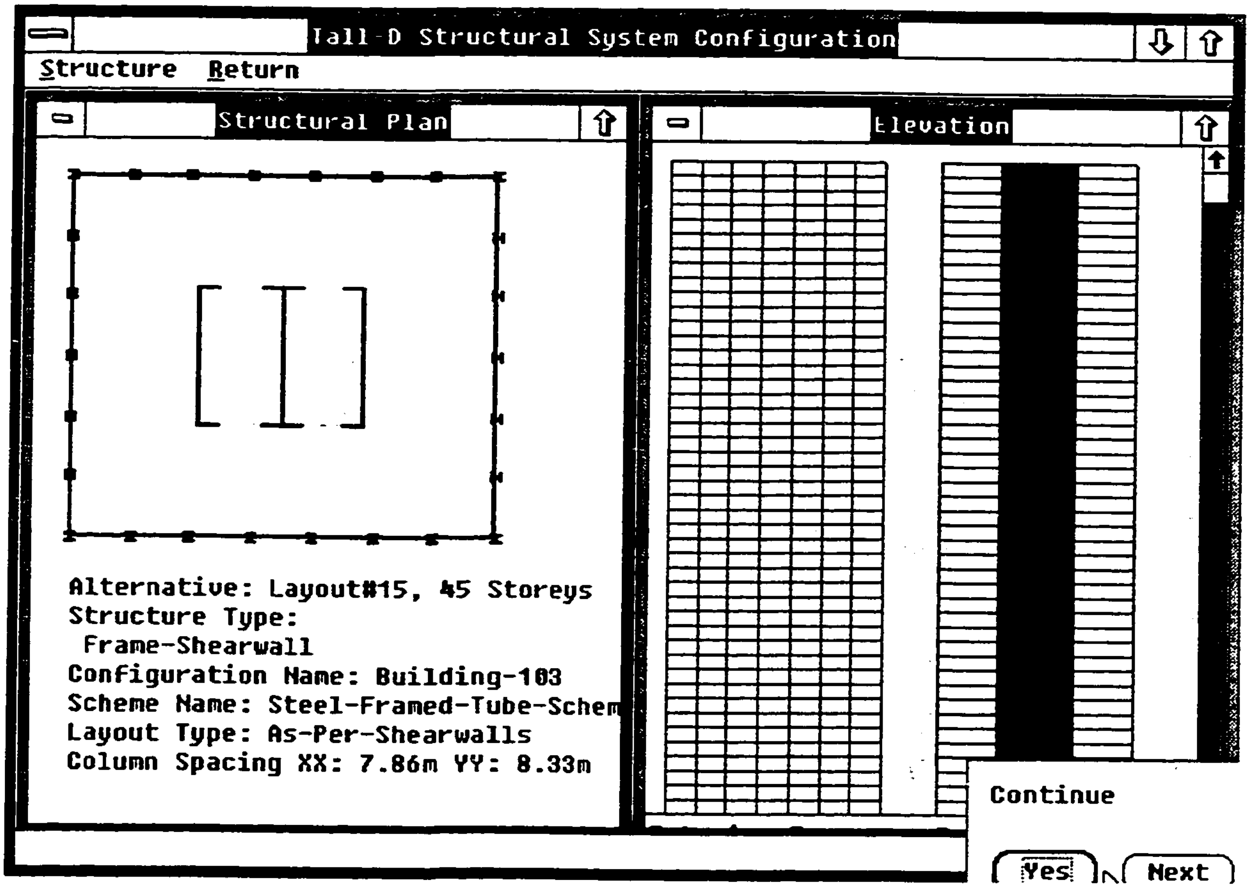


Fig. 6.3.11 Plan of a steel tubular structure alternative for Layout#15 of Design Case 3.

Tall-D core which is proportioned in the ratio of 2:1. Therefore the cross-sectional areas of both designs are closer than the thicknesses or number of walls would suggest.

The steel column sizes obtained with Tall-D seem somewhat smaller in size than used in the actual case. The Tall-D sections are based on reduced tributary area as permitted by the codes. The effect of the lateral loads was implemented by increasing the tributary loads by fifty percent. It is however evident that the Tall-D steel column designs may be somewhat under-designed. Consequently the weight of steel quantity estimated by Tall-D is somewhat lower than what it actually is. However the tonnage used on a relative basis can be a useful tool as illustrated below. Comparing the steel tonnage for the columns, the tonnage for the configuration closest to the actual (spans of 7.9m and 8.3m in Table 6.6) is 297 metric tonnes. The other tonnages being 432, 351 and 594. See Appendix D-3 pages D3-11 and D3-12 for the steel weights in kilograms for the alternative column spacings in this steel framed shearwall layout. Tall-D specifies the steel strength as 220MPa as opposed to the higher strength steel used in the project for columns. But for an implementation oversight in the steel strength selection function, Tall-D should specify 345MPa steel as required by the chart used in steel column sizing. For a range of higher tributary area, the charts (Allen and Iano 1989) used in Tall-D, require the use of higher grade 345 MPa steel which is not reflected in Tall-D implementation. Neglecting the material strength, the columns sizes at the different levels are only marginally under-designed as can be seen in Table 6.6.

The gravity system details are summarised in Table D3 in appendix (p. D3-14) including only the top five alternatives for each of the two LLS schemes (the concrete frame shearwall scheme and the steel tube with concrete shearwall scheme). The composite-truss with deck slab that was used in the project was recommended by Tall-D and also ranked first for the steel framed tube system (see page D2-14, lower half of the table). Truss beams are also excellent to embed service ducts with ease compared to many other steel gravity systems. The actual total depth of this truss-beam is 920mm which compares to 750mm deep joist plus 70mm steel deck plus a similar amount/depth of concrete topping bringing the total Tall-D floor system depth close to 900mm. However, the Tall-D span is larger, 15.3m and 16.8m compared to 13.4m in actual. Therefore, as the expert noted the depth is somewhat inadequate from floor vibrations point of view but good from strength point of view.

As the expert read through the GLS alternatives and their dimensions, some systems are to be discarded and some other are to be increased in depth for serviceability considerations. Hollow core slab and double Tee beam systems are suggested to be discarded possibly due to the frame being of steel. Composite steel beam with deck is considered adequate from strength point of view but may have serviceability problems such as vibrations. Composite stub girder system is judged to be under designed with a depth of 630mm and 800mm to 900mm is suggested as appropriate. Castellated beams suggested by Tall-D with a depth of 788mm is suggested to be used only as girders, not as beams evidently due to vibration problems.

Two other structural systems are identified by Tall-D as suitable for use in these buildings, but did not complete the design due to the incomplete state of their implementation. These two are framed end channel scheme and belt truss with braced frame scheme.

The number of mechanical floors is also low - two compared to the four in the actual case. No particular reason is seen as the previous data from sources such as Guise (1990) suggested the values generated by Tall-D were keeping in order with similar designs. Since Tall-D uses constant building core size for preliminary design purposes, when it in fact reduces in size as some elevators drop off with height, the total area designed for in the building is however adequate for additional floor or two of service area. This can be confirmed by examining the gross and net floor area of the as-built and test-case.

Another part of the structural system is the set of spandrel beams that are on the perimeter of the building, participating in the lateral load resistance. The actual building had a section size of WSP1000@130 along the building perimeter between the 18th floor and 39th floors. The reason being that the perimeter frame in this range of floors participates in the lateral load resistance and are hence moment-connected. Elsewhere on the perimeter the depth of beams is in the range 410mm to 530mm and are not moment-connected. Tall-D in this context however restricts itself to the one important design variable, the depth of the gravity load-resisting system (the beams that span most and have maximum depth), which it is due to the repetitive nature and number of floors. The Tall-D depth of 900mm is commensurate with practical values from the point of view of strength although from serviceability point of view the beam and slab depths may need to be augmented marginally. Such an opinion was in-fact expressed by the structural engineering expert.

The gravity load-resisting system alternatives consist of different steel and concrete systems. A composite truss deck-slab of overall depth 850mm with a corrugated steel deck 76mm deep is used in the test-case. Tall-D includes among its alternative solutions the same type of deck with a smaller 750mm overall depth with 70mm deep corrugated

Table 6.6 Comparison of as-built 1000 de La Gauchetière building with Tall-D design.

Building Parameter		As Built		Tall-D (Layout#15)	
Plan dimensions		51m x 42m (12th floor & above) 63m x 42m (below 12^{th} floor)		55m x 50m	
Core dimensions		36m x 12m		21.3m x 19.4m	
Storeys (+ mechanical levels)		43 (+ 4)		43 (+ 2)	
Floor area (Gross:: Net)		n/a :: 100000 (m^2)		123750 :: 100482 (m^2)	
Cost of building ($ millions)		125 for tower*		111	
Lateral-load resisting system		Steel Frame-Shearwall		Steel Framed-Tube†	
Material grade of columns		300 Mpa		220MPa/345Mpa	
		Front:xx	*Side:yy*	*Front:xx*	*Side:yy*
Column spacing on perimeter		6.0m	6.0m	7.86m	8.38m
		Section@KG/m		*Nominal Size mm x mm*‡	
Column sizes on building perimeter (mxm)	Ground Level	BH600@936	WWF550@620	W610x325	W610x325
	Levels 8-10	WWF600@551	WWF450@409	W610x325	W610x325
	Levels 20-22	WWF500@343	WWF450@409	W460x280	W460x280
	Levels 29-31	WWF450@228	WWF400@362	W360x360	W360x360
	Levels 38-40	W310@107	WWF400@362	W310x310	W310x310
	Levels 45-47	W200@49	WWF200@49	W250x250	W250x250
Number of shearwalls		6		3	
Shearwall concrete, size and thickness		55MPa; 2#12.65m ; 600mm thick 4#12.65m; 400mm thick		55MPa; 2#15.7m; 675mm thick 1#15.7m; 600mm thick	
Gravity-load resisting system		Composite-truss with steel deck slab		Composite-truss with steel deck slab§	
Clear span (front/back & sides)		13.4m$\Updownarrow$ 7.5m$\Leftrightarrow$		15.3m$\Updownarrow$ 16.8m$\Leftrightarrow$	
Depth of gravity system beams		920mm; including 150mm deck slab (72mm deck + topping)		750mm + 70mm steel deck + concrete topping	

Notes:

* Additional costs of $30 millions for under-ground lateral earth pressure resisting system and $25 millions were spent for the plaza as per the engineer.

† Other alternatives generated are steel concrete frame-shearwall, steel braced-frame, belt-truss braced frame and framed end channel.

‡ Initial sizing routines select steel sections in nominal section dimensions; this facilitates precise weight designation at detailed design stage;

§ Composite-truss with deck slab is ranked first for this alternative (see pg. D3-14).

steel deck as shown in Appendix D-3. According to the expert, the overall depth in the latter case is inadequate.

6. 3. 7 Questionnaire and Expert Comments

A questionnaire with 40 different queries categorised under six headings is used to elicit an assessment of Tall-D from the experts. Table 6.7 shows a comparison of the scores on questionnaire by the two experts, based on a scale of 1 to 5 with 1 being "strongly disagree" and 5 being "strongly agree". A few of the features such as member scaling for GLS being under-designed invited a grading of 2 from the engineer (see section IIb, Table 6.7). Otherwise the system was graded favourably in all other categories. On a few items the two experts differed significantly in their assessment. On the very first item - relevance of floor plans generated - the architect scores 5 versus a 3 from the engineer. Similarly on page 2 of the same table, the item - display of results - got 2 from the engineer and 4 from the architect. Apart from perceptions, one reason for this is the architect expert performed two designs - one of moderate height and the other much taller. Therefore he was exposed to a larger variety of configurations and corresponding number of text screens, giving a better view of the same. Barring a few items out of some 40, there is general agreement in their assessment. Both experts give 3 for approximate costs indicating it needs improvement and are highly enthusiastic of the utility of Tall-D. Table 6.8 shows the average scores in each category that Tall-D is evaluated in. The averages show the engineer scoring from a low of 3.0 in cost and quantity to a high of 4.6 in utility of Tall-D categories. The architect scored a low of 3.6 on design process issues and user interface with the high being again utility of Tall-D with a score of 4.4. Incidentally the "Utility of Tall-D" category consists of items such as useful to perform preliminary design (both score 5), useful to explore design possibilities (again both score 5) and useful tool for multistorey building design (scores of 5 and 4). This clearly demonstrates a principal aim of this work is realised which is the development of a tool for preliminary design. Since an architect along with an engineer considered Tall-D from their points of view and responded so well, it can reasonably be said that domain integration has been successfully demonstrated.

Table 6.7 Comparison of the Tall-D Evaluation Scores by Expert Structural Engineer and Expert Architect.

Tall-D Evaluation Criteria	(5) Strongly Agree (1) Strongly Disagree	
	Engineer	Architect
I. ***Architectural Considerations (flexibility etc.)***		
Relevance of floor plans generated	1 2 3 4 5	1 2 3 4 5
Core location(s) appropriate	1 2 3 4 5	1 2 3 4 5
Relevance of column layouts generated	1 2 3 4 5	1 2 3 4 5
Ranking of initial floor plans	1 2 3 4 5	1 2 3 4 5
II. ***Preliminary Structural Design Considerations.***	.	
IIa. ***Geometric Configuration of Lateral Load-Resisting System (LLS)***		
Exploration/Generation of possible LLS alternatives	1 2 3 4 5	1 2 3 4 5
Compatibility of LLS with initial floor layout		
Of plans with rigid-frames	1 2 3 4 5	1 2 3 4 5
Of plans with braced-frames	1 2 3 4 5	1 2 3 4 5
Of plans with shearwalls	1 2 3 4 5	1 2 3 4 5
Of plans with tubular structure	1 2 3 4 5	1 2 3 4 5
Concrete Column Sizing	1 2 3 4 5	1 2 3 4 5
Steel Column Sizing	1 2 3 4 5	1 2 3 4 5
Concrete Shearwall Sizing	1 2 3 4 5	1 2 3 4 5
IIb. ***Geometric configuration of Gravity Load-Resisting System (GLS)***		
Exploration/Generation of Relevant alternatives	1 2 3 4 5	1 2 3 4 5
Concrete beam sizing	1 2 3 4 5	1 2 3 4 5
Steel beam sizing (rolled sections, stub girders,		
castellated, haunch, tapered beams, joist/truss etc.)	1 2 3 4 5	1 2 3 4 5
Concrete slab sizing (flat-slab, waffle, joists, oneway and twoway slabs etc.)	1 2 3 4 5	1 2 3 4 5
Steel deck-slab sizing	1 2 3 4 5	1 2 3 4 5

Table 6.7 (continued) Comparison of the Tall-D Evaluation Scores by Expert Structural Engineer and Expert Architect.

Tall-D Evaluation Criteria	Engineer	Architect
III ***Design Process Issues and User Interface for Tall-D***		
Consideration of relevant constraints	1 2 3 4 5	1 2 3 4 5
Ability to redefine criteria	1 2 3 4 5	1 2 3 4 5
Ability to go back to generate different schemes	1 2 3 4 5	1 2 3 4 5
Sketch Floor Layout Graphical Display	1 2 3 4 5	1 2 3 4 5
Structural Elevation Graphical Display	1 2 3 4 5	1 2 3 4 5
Display of Results - Text Screens	1 2 3 4 5	1 2 3 4 5
Explanation of Results	1 2 3 4 5	1 2 3 4 5
Clarity of Menus and prompts	1 2 3 4 5	1 2 3 4 5
IV ***Approximate Analysis of frames for lateral loads***		
Rigid Frame system	1 2 3 4 5	1 2 3 4 5
Braced frame system	1 2 3 4 5	1 2 3 4 5
Frame-Shearwall system	1 2 3 4 5	1 2 3 4 5
Tubular frame system	1 2 3 4 5	1 2 3 4 5
V ***Cost and Quantity Estimation***		
Approximation of overall project cost	1 2 3 4 5	1 2 3 4 5
Relative structural cost of vertical structuarl system	1 2 3 4 5	1 2 3 4 5
Relative structural cost of floor system	1 2 3 4 5	1 2 3 4 5
Structural quantities estimation	1 2 3 4 5	1 2 3 4 5
VI ***Utility of Tall-D***		
Overall concept of System	1 2 3 4 5	1 2 3 4 5
Useful to perform preliminary design	1 2 3 4 5	1 2 3 4 5
Useful tool for multistorey buildings design	1 2 3 4 5	1 2 3 4 5
Useful to explore design possibilities	1 2 3 4 5	1 2 3 4 5
Useful as learning tool for students	1 2 3 4 5	1 2 3 4 5
Overall performance	1 2 3 4 5	1 2 3 4 5

Table 6.9 shows the written comments that were submitted in addition to the comments and observations gleaned during the design sessions. While the architect was enthusiastic in pointing to the possibility of a commercial application, the engineer was enthusiastic of the potential of Tall-D as a fully developed tool for preliminary design.

6. 3. 8 Review, Limitations and Conclusions

The validation of Tall-D was performed in two stages. The first was an incremental testing by the author as the system was being developed using many routine methods like tracers, browsers and inspectors for rules, frames, instances and functions. The other more general validation was to ask external experts to use the system not only to examine the general features of Tall-D, but also to perform preliminary design for some of the tall building projects they are familiar with. Tall-D was well received by the two experts who were involved at this step of the validation process as presented in the previous section (6.3.7). Though there were no totally outrageous design alternatives proposed it is relevant to mention limitations found in Tall-D.

Here follows a few weaknesses in Tall-D, some of which have been mentioned by the author elsewhere or were pointed out by the experts. Both experts considered that the cost was under estimated to the extent of 10 to 15 percent. One reason is that the latter two of the test cases were considered triple A category - the highest in office rental properties. Due to this fact the cost of construction of such facilities is somewhat above the average cost index values used in Tall-D implementation. Perhaps a correction for this aspect in Tall-D would make the cost estimation in Tall-D more realistic.

The architect pointed out that the core size was large for the Place du Canada building. It was 20% of the gross floor area when Tall-D was tested. The expert suggested 15% as more appropriate and even 10% for buildings smaller than that. Place du Canada being much smaller than the other two test cases, it was an appropriate observation. However for the larger two test cases a 20% core space was considered reasonable. The 1000 de La Gauchetière building and the IBM-Marathon building had a core shape that

Table 6.8 Average scores by expert engineer and expert architect in Tall-D evaluation.

Tall-D Feature Classification	Expert Engineer	Expert Architect
	(1)Strongly Disagree (5)Strongly Agree	
I. Architectural Considerations	3.5	4.0
II. Preliminary Structural Design Considerations.		
a. Geometric Configuration of Lateral Load-Resisting System (LLS)	3.6	4.0
b. Geometric Configuration of Gravity Load-Resisting System (GLS)	3.2	4.0
III. Design Process Issues and User Interface for Tall-D	3.5	3.6
IV. Approximate analysis of frames for lateral loads	3.8	4.0
V. Cost and Quantity Estimation	3.0	4.0
VI. Utility of Tall-D	4.6	4.4
VII. Overall performance	4.0	4.0

Table 6.9 Written comments of experts further to evaluation of Tall-D system

Comments by Mr. Serge Vézina. Ing., Ordre des ingénieures du Québec member.
The Tall-D system, as it stands, has the potential of becoming a valuable tool for the practicing engineer at the conceptual stage of a multistory building project. It provides schematic representations of various layouts of floor plans and structural elevations as well as realistic cost estimates for the different alternatives analyzed. Furthermore, the Tall-D system incorporates, in a very practical manner, several aspects of the design of multistory buildings. The hierarchical generate-test technique used in the computer-based system as well as the integration of approximate methods of structural analysis in the KBES, truly forms the basis of what could become a fully integrated system for the conceptual design of multistory buildings. In that sense, Mr. Ravi Mathi did achieve his goal of designing an efficient and practical knowledge-based system that integrates the architectural and engineering constraints relevant to the design of multistory buildings.
Comments by Mr. Antonio Niro, Ordre des architectes du Québec member.
As discussed during the presentation, a strong possibility of practical application would exist, in my opinion, except for the fact that the market for such a product is quite limited.

was more elongated than Tall-D core shapes. This resulted in a slightly smaller clear span in the XX direction and larger clear span in the YY direction compared to the actual designs.

A noticeable weakness in Tall-D member sizing is related to steel beams located on the building perimeter and moment-connected so as to participate in lateral load resistance. The initial sizing of beams in Tall-D is oriented to finding the depth of the gravity system that applies to the entire floor. This depth of gravity system beam is based on the span, which results in a lower depth when compared to those beams that participate in lateral load resistance, like the ones on perimeter which are moment-connected. Unlike the strategy in columns where the lateral load is accounted for by designing with augmented axial loads, no such step is implemented in the design of moment-connected beams. The result was the seemingly under-designed beams on the perimeter compared to the perimeter beams in the third design case (1000 de La Gauchetière) where the perimeter beams in the middle third of building height are moment-connected while the rest of the perimeter frame is shear connected. In the corresponding Tall-D design, lateral loads were considered to be resisted by shearwalls with minimal contribution from perimeter frames. However a simple correction to the Tall-D method can be adopted as an increase in perimeter beam depth to account for the lateral load moments in a frame-shearwall interaction structure. If a more precise solution is desired, the portal method of analysis can be used for lateral load analysis before member sizing is performed.

Gravity system sizing was in some cases under-designed. Designs produced by Tall-D were considered by the experts as adequate from the strength point of view but could be inadequate from the serviceability (structural) point of view. To improve that aspect, a review of the knowledge base and the charts used for initial sizing can be of assistance. A last aspect that can be improved in Tall-D is the display of gravity system alternatives in the browser. However owing to the increasing scope of such programming tasks not directly related to the building design content of the knowledge base, such user interface issue was given a low priority in development.

These are primarily some of the issues that were discussed during the Tall-D validation process in addition to what is already presented in the questionnaire. Other issues of scope for further research are mentioned in the next chapter, section 7.4. The next chapter, the last, presents an extended summary of the entire work noting the contribution to state-of-the-art in all related areas of this research, the limitations of the current tool Tall-D as well as scope for further work in this area, and specifically in Tall-D like tools.

CHAPTER 7

Summary and Conclusions

The work described in this thesis represents a significant original contribution in identifying, acquiring, synthesising and organising design knowledge related to tall buildings. Due to the non-formalised nature of preliminary design, no computer-based tools could be developed using common procedural programming methods. The preliminary design stage also has the maximum ability to impact the final cost of the building project. In the absence of computer-based tools and also due to resource and time constraints, the current preliminary design tasks are mostly restricted to exploring one or two design alternatives, often proceeding with a less than efficient solution. Therefore a computer-based design tool that can be used by designers at the initial stages of a project to explore many design alternatives on a comparative basis will, lead to better and more economical designs than those produced without such a design tool. Preliminary design needs the consideration of all relevant parameters affecting the product designed. For multistorey buildings, the architectural and structural design considerations are closely tied together and in this work an integration of the two is demonstrated. Architectural planning considerations for office buildings and feasible structural solutions were incorporated in a knowledge-base. A model for the design system is developed that incorporates the building design knowledge in declarative form which is then used to selectively generate and evaluate integrated building design solutions. Thus, part of this work is the implementation of a knowledge-based system - Tall-D - using a development tool. Then, a methodology is devised for the testing of Tall-D under practical conditions. External industry experts were invited to use their completed project designs and evaluate the corresponding designs performed by Tall-D and answer a comprehensive questionnaire. The result is a favourable endorsement of the many facets of the current work as well as that of Tall-D system performance and features. The overall thrust of this work has been to research the area of the preliminary design process with a view to originating prospective computer-based tools for the preliminary design phase in engineering.

The knowledge-based design system, Tall-D, is capable of assembling overall building configurations and compatible structural systems for multistorey office buildings. Alternative building spatial and physical configurations are generated with consideration of planning aspects as well as structural efficiency. Alternative floor plans and shapes are evaluated against a set of performance criteria. The result is a group of different building overall configurations that are ranked according to the designer's preferences. This is followed by the generation of structural system layout alternatives that are compatible with the building configurations. The structure geometry and initial structural element sizes are generated. The main research contributions and scope for further research are presented in the next two sections.

7. 1. Main Research Contributions

The general area of major contribution of this work is Computer-Aided Building Design (CABD). Traditionally computer-based systems have been used to perform number crunching tasks to solve precisely defined calculations. Preliminary design on the other hand has had little application of computers. In preliminary design stages, exploration of relevant alternatives is an important task. This research has effectively contributed in that area by developing the paradigm to systematically generate and communicate the design space. This work has successfully identified a solution for preliminary design for the domain of multistorey building design. The following sub-sections describe the contributions of this work.

7. 1. 1 Preliminary Design Domain Knowledge for Buildings

One of the contributions of this work is the identification and collation of the domain knowledge for preliminary design of buildings. Preliminary design is an area of activity in which expert engineers and architects are involved. To identify the knowledge-base it is required to study the type of traditional design tasks at the preliminary design stage and also identify areas where the computer can extend and aid these design tasks. It is known that preliminary design is largely heuristic. Many experts do not generate a comprehensive set of alternative solutions, instead they depend on their experience to restrict their design choices to a few. However, designers would like to explore many, relevant alternatives if there is a tool available that will preclude substantial additional time and resource demands on them. This work enables such a method of preliminary design by considering many, but relevant alternatives in a given design context.

The contribution in this area is therefore in the collection and collation of knowledge for the generation and evaluation of alternatives as follows. The design knowledge for the generation of overall building configuration alternatives is identified. Information on factors such as service areas, zoning laws, building proportions, circulation, egress distances, rentability, energy efficiency etc. has been collected. The knowledge-base developed out of this enable Tall-D to generate many floor plans and building elevations that combine to form an overall building configuration.

Information on the generation of alternative vertical and horizontal structural systems for any given floor plan is the other major component of the domain knowledge. The various prevalent structural layout footprints have been studied. The resulting knowledge-base enables the structural module of Tall-D to generate the layout of structural elements such as shearcores, rigid frames and gravity load resisting systems. A collection of approximate methods of sizing the many different structural elements has also been put together. Many charts and tables for this purpose were identified. Although many approximate methods of analysis were studied for a more rigorous analysis of the different structural systems, one method with wide applicability (portal method) is implemented in Tall-D as a stand-alone module.

7. 1. 2 Formulation of Design Process for Preliminary Design

Another primary contribution of the current work is to formulate the non-formalised preliminary design process suitable for a computer based system. Having identified the domain knowledge as briefly described in the previous section, two aspects of this task can be distinguished.

First, for the preliminary design process, an artificial intelligence technique - hierarchical generate-test paradigm - is adapted for implementation. The paradigm to generate, store and evaluate multiple and simultaneous alternatives is unique. The representation of alternatives arising out of a common root by variation of a design parameter has been addressed. The methodology of spatial reasoning which assumes importance in engineering applications due to the physical nature of the design objects that are dealt with in areas such as structures and buildings is another important aspect. Rather than combinatorial, exhaustive generation and evaluation of alternatives, an approach of selective generation based on design context is achieved here by the use of the KBES approach. The designer also has control in choosing and selecting floor plans to generate complete structural design.

Next, the product - the building - is organised. At the architectural design stage formal descriptions of plan layouts with various core locations are identified. At the structural level various structural system layouts are also identified and semi-formal definitions to describe them are developed. This provides a transition from architectural/planning layouts to the structural layouts.

An object-based representation is adapted to represent the building components with their elaborate design parameters and used for run-time generation of alternatives. Due to the requirement of multiple instantiation of alternatives, a specific method of representation of alternatives is developed. The finer details of an alternative are developed with the introduction of more and more design details to the alternatives at the previous step (or level) in the generation process. When more than one possibility of that design detail is encountered, it represents a branch off in the generation process. Where the basic nature of the alternatives remains the same, such alternatives are represented by the same object instance bit, with multiple-value slots for the property in question. An example is a 'frame shearwall' structural scheme with alternative column spacing on the perimeter, that is in practice evaluated as an alternative, is represented by Tall-D using a single instance for all perimeter column spacings. The advantage is economy in representing alternatives and clustering of alternatives with many common features.

While current systems generally consider several design details as given to start with, Tall-D initiates building design with very little data and gradually generates complete solutions. Multiple criteria for evaluation at the preliminary design stage are entered in the system and provide a realistic approach for selecting overall configurations.

Good designs are judged by their overall cost. At the preliminary design stage it is one of the main criteria used to compare alternatives. In practice, designers evaluate relevant alternatives by using conventional programs for costing. This is feasible only when preliminary design alternatives are available. In generating early design alternatives however, one has to rely on prior knowledge to initiate such generation. A system such as Tall-D greatly enhance this phase of design by generating many relevant alternatives with associated cost and quantities that serve well for comparison purposes.

The preliminary design process should provide a transition to a more detailed design. The ability of Tall-D, to generate the final structural geometry through a neutral text file that can be imported by many CAD programs, provides a level of interface to the next phase of design.

An aspect of preliminary design is the integration between domains. There are new developments driven in part by software vendors that are aimed at exchanging or sharing data between different programs and even different types of computer systems. There are also developments, driven by the engineering design community, that are aimed at the definition of standardised product data models with a view to enabling and sharing design data. However there is no effort to formalise the inter-play of different domain knowledge areas that influence building design. Tall-D has addressed this issue in building design integration at the domain level by developing and using architectural and structural engineering domain knowledge modules. It thereby demonstrates the essential nature of integration of different viewpoints of a design task.

The contribution by way of formulating the many issues in the preliminary design process with a view to generating a formal design knowledge-base was discussed. The next section describes the contribution in the actual development and implementation of such a knowledge-base.

7. 1. 3 Formalisation and Implementation of Design Knowledge

The domain knowledge and the preliminary design process discussed in the previous two sections need a test-bed in the form of a KBS. The formalised knowledge-base for the design of lateral and gravity systems for tall buildings as well as overall design considerations in architectural planning is an important contribution of this work.

A comprehensive declarative model for a building and its component parts, complete with semantic relationships is developed using object-based technology. This enabled an efficient knowledge representation in conjunction with production rules. The latter is used to represent the procedural knowledge in preliminary design. Together, the hybrid knowledge representation along with many routine procedural functions implementing heuristic charts and tables result in a functional test-bed KBS called Tall-D. Tall-D system is used on a total of four demonstrative examples, three of them being actual buildings used by industry in Tall-D validation. The result is a significant contribution in advancing the state-of-the-art in formalisation, implementation and validation of preliminary design as a computer-based design activity. Thus, beginning with understanding of the domain to the development of a design tool, this work represents advancement in knowledge in the area of computer-aided building design.

Different levels of abstraction in knowledge representation as well as a multiple generate-test procedure are found to constitute a practical step-by-step approach with building design problems. For efficient design decision-making at the overall level, it is essential to reduce the search space to a limited set of alternatives.

Knowledge representation of design artifacts by frames (or objects) and instances having associated functions like demons, handlers, constraints and inheritance, combined with production rules, facilitate the development of a KBES incorporating the multiple generate-test procedure. A combination of production rule system and object-oriented programming is very well suited for large systems such as for integrated design. The production rules help carry-out the inferential process. The message passing and knowledge-representation facilitated by the use of frames help the maintenance of a design description with a far less number of rules than would be required without frames. The facility to invoke procedural routines from within both the rules and the frames has also been widely used to carry-out many different operations in the KBS. It is also felt necessary to arrange rules into modules that represent specialised tasks in building design. As a result of this work is it observed that there is a need for all three types of computing, if an efficient KBS is to developed - both from implementation and domain point of view.

The use of a graphical interface in Tall-D greatly increases the utility in such engineering design systems as demonstrated by the current work. It is easy and advantageous to check the physical representation of design decisions on-screen, as designers routinely do in practise. Previous research in preliminary design systems is generally lacking such interfaces.

The present study has demonstrated that preliminary design as well as multi-disciplinary design decisions can be efficiently supported by KBES techniques. The difficult computational task of incorporating experiential knowledge that is essential at the preliminary design stage is addressed effectively. An alternative to traditional procedural methods of computation is found in the declarative KBES techniques to facilitate preliminary design. The KBES developed is used as a tool for investigating the influence of multiple sources of information on the development of overall building configurations and compatible structural systems.

7. 1. 4 Validation Approach for Preliminary Design Systems

A unique aspect of this work is the method adopted for the system validation. Though no experts are involved in the development of Tall-D, there have not been many studies reported where completed projects have been used with the experts themselves participating in the validation process.

Three completed projects were furnished by the experts and Tall-D performance on those designs were observed. This observation by the experts formed the basis for second stage of Tall-D evaluation. The experts answered a questionnaire with about forty Tall-D performance-related attributes. The questionnaire served to evaluate Tall-D performance on a scale of 1 to 5. The two expert evaluations (scoring pattern) are closely correlated and quite positive for the most part.

In the process of evaluating the system and answering the questionnaire the two experts also underscored the vitality of design aids like Tall-D. By their responses to the five points in the final section of the questionnaire 'Utility of Tall-D', with an average score of 4.5 out of a possible maximum of 5, they have endorsed the view that a solution for preliminary design issues in multistorey buildings has been effectively addressed.

Giving this tool to the designers in the industry, and evaluating and using the feedback to improve the system design can not only help the engineers in preliminary design, it could improve the market and awareness for such systems.

7. 2. Scope for Further Work

Though this work has clearly demonstrated the utility and progress towards computer-based preliminary design tools, there is a large scope for further research in preliminary design and integrated design areas. The simplest and direct extensions to the current work are presented followed by more general and more elaborate projects.

7. 2. 1 Enhanced Building Design Knowledge-base

In order to make a system like Tall-D much more useful, the current work can be extended with respect to the following areas that directly impact the building design knowledge content. Some of the features described as possible enhancements can also be seen as limitations of Tall-D.

a) Architectural Domain

A more detailed massing and space allocation module can be developed. Architects prefer using grids with unit area usually 5'x5'(1.52mx1.52m) so that occupant space planning is simplified. The building plan layout is placed on the grid lines. This latter functionality can be added to Tall-D by modifying the generation of initial layouts using overall dimensions that are multiples of this grid spacing.

Explicit consideration of egress and fire safety considerations can be implemented so that variable criteria can be specified to reflect local regulations. Currently these concerns are incorporated in the maximum permitted plan dimensions. Similarly zoning regulations may vary or special exemptions may be granted for special project. Again, the floor area ratio and similar zoning regulations may be implemented as adjustable parameters. Similarly, instead of considering all storeys of uniform height, the storey height of the mezzanine level may implemented as different from the rest.

Occupancy types other than 'office' can be considered to widen the use of Tall-D like systems for design of all types of buildings. For example, high-rise residential buildings have different plans and consequently different structural layouts. The current knowledge-base can be enhanced to support additional building occupancies.

The knowledge-base can be augmented to support less common plan layouts for multistorey buildings so that Tall-D can perform well on special cases on par with the common plan layout types. Examples are: plan with a detached core and a plan with two end cores. Even entirely new configurations such as circular or plans with curvilinear outlines could be introduced in Tall-D.

b) Structural Domain

The knowledge-base currently is heavily tilted in favour of buildings up to thirty stories. Not all possible structural system details for buildings above that range are generated by Tall-D. This aspect could be improved by enhancing the structural knowledge-base by including structural systems such as, belt-truss, tube-in-tube and framed end channel types among others. Also composite material construction can be considered identifying where they are more economical than steel or concrete alone.

Floor layouts can be refined by considering variation along the height of the building instead of a typical floor considered by Tall-D. In taller buildings elevator banks drop off creating decreasing core sizes and more occupant space at the higher floors. More detailed design of core proportions, layout and core structure will improve the designs produced by Tall-D and make them closer to detailed design.

Tall-D users have to use the quantities for the lateral load resisting system and the depth of the gravity system as a means of prioritising the alternatives generated. Due to computing hardware limitations it is not possible to generate all the alternatives for all structural system schemes simultaneously. Rather one is forced to select two at a time when structural quantities are also generated. Currently the evaluation of structural alternatives is based on relative quantities and the comparison is left to the user. The structural evaluation can be enhanced by KBS tool upgrade as well as hardware upgrade so that all the structural alternatives' quantities can be generated simultaneously to facilitate structural evaluation based on quantities, footprint, drift, gravity system depth, in-place costs, constructibility and so on.

As mentioned, upgrades to GoldWorksII, the KBS tool used, as well as hardware upgrades will make Tall-D system a more useful tool for architects and engineers of tall buildings. With a newer version of the tool, some of the weaknesses of Tall-D due to an upper limit on the ability of GoldWorksII to use memory, can be overcome so that the designer can explore all feasible structural alternatives at once. Currently only user selected configurations are processed for structural member sizing. Lateral load resisting rules sets have to be unloaded from memory before compatible gravity load resisting systems can be generated by Tall-D. If the user then wants to regenerate any LLS, Tall-D LLS rulesets have to be reintroduced (or loaded) into the KBS environment.

Integration between design and analysis is another area where there is scope for further work. The integration of the portal method implemented in Tall-D which is currently stand alone, with the rest of Tall-D system will improve the structural design of girders subjected to wind moments. Further work could also employ the portal method to generate complementary solution to obtain a total solution for the different structural systems by approximate methods as listed in Table 5.3.

It would be a useful exercise to look at Tall-D output as experimental data to analyze and try to infer generic conclusions for the design of tall buildings.

c) Mechanical, HVAC and Elevators

Inclusion of other domains to enhance preliminary building design decisions by systems like Tall-D is an area for further work. Beginning with mechanical systems like heating, ventilation and air-conditioning (HVAC), vertical and horizontal service ducts and vertical transportation. There is need to identify all the aspects of these systems that influence preliminary design of buildings and then to organise and formulate the knowledge-base. At a global level, for example, the building orientation and plan proportions have an effect on the energy use. At a more intricate level the type and depth of gravity system is linked to HVAC system design or vice-versa. Elevator and Vertical transportation system effect the Core structure and shearwall dropoff. Precise core layout can be done only in conjunction with vertical transportation system design. Inclusion of such criteria as these and also in increasing detail will gradually take Tall-D where it can optionally perform tasks that are generally undertaken at the detailed design stage. The result could be another insight in integrated design at the preliminary stage if the new system can be successfully demonstrated with external experts serving to validate the system.

d) Cost Estimation

The current cost estimation in Tall-D is an order of magnitude estimate. Preliminary cost estimation could be accurate to within 25 percent of the actual cost. Therefore it is reliable only to use relative cost or quantities as a basis for cost comparison between alternatives as is done in Tall-D. There is scope to enhance the functionality to a more precise cost estimation. First the quantities estimation function needs to be completed in Tall-D such as for gravity structural systems. Then, interface with current on-line construction industry databases and facilitate the use of the cost database. An improved cost reporting interface is also needed to make this a useful feature.

7. 2. 2 Enhanced Integrated Design

The current work addressed the issue of integration in the context of preliminary building design and also in the context of multiple domains (architectural and structural in Tall-D) for knowledge-base systems. Thus, only a limited aspect of the global issue of 'integration' in computer-aided engineering (CAE) is addressed. Due to the islands of automation current in the many engineering disciplines, a multifaceted integration problem exists.

An aspect of integration is directly related to data and information. There is loss of productivity when at different stages along the life cycle of a facility (design, construction, operation and maintenance) essentially the same data, albeit in different format is input and used in different programs. Such fragmented and replicated data for different purposes (and different programs) introduces scope for errors in data translation and communication of information. It also greatly restricts the potential of data and information use in organisations. Different programs and different activities in facilities design, construction and management could greatly benefit if all the different computer applications were developed such that they could all use a common data source. Development of standards for data sharing and data exchange is one approach that has been prevalent for some time. Research on how CAE/CAD/CABD applications can fully benefit from such approaches has to be performed. An approach to integration through common data standards is also possible. If design data, both geometric and engineering knowledge (intent implied or otherwise), is carried from the conceptual stage right through to the building, operation and maintenance of facilities, we would then have reached the full potential of computer-based integration.

The other aspect of integration is related to domain knowledge. Any practical engineering design problem encompasses many disciplines. Therefore it is inevitable that multiple knowledge sources and interacting knowledge sources be considered. Tall-D has considered two closely related domains in the preliminary design of tall buildings. This aspect can be examined afresh using the cooperating expert systems paradigm.

7. 2. 3 Apply to New Areas of Building Design

The building envelope and structure of the facade have a high degree of interaction. Design considerations affecting both subsystems are aesthetics, energy consumption, initial cost and maintenance cost in addition to size and spacing of structural elements.

Thus building structure and envelope exemplify a level of integration that could be explored to further the use and development of integrated design systems. Similarly construction technology and advances in materials can often tilt decisions in favour of specific options. Such knowledge can be included as part of the integrated approach. Preliminary design could be expanded to include such considerations as soil conditions, the design of substructure, effect of neighbouring buildings on wind loading, interaction of structure and envelope, HVAC, lighting and an enhanced cost engineering component to account for the above additions. Interface to a comprehensive energy model, would help in the estimation of energy cost. Such information could be used in the life cycle cost analysis. The model for integrated design could thus be enriched by including more domains along with the current areas of architecture and structure.

The computer-based model for integrated design could be extended to include detailed design aspects. Detailed design of multistorey buildings has levels of complexity that could benefit from KBES technology.

Preliminary design has features that are unique to that phase of design. An important feature is the method of generating and reasoning with a large number of alternatives. Consequently, the knowledge representation and the economy of such representation becomes important. The work done in developing Tall-D could be a basis for further research in this aspect of preliminary design. Knowledge representation of dynamically generated alternatives, that are incrementally different from their previous ones, leading to a combinatorial explosion, is a challenge. KBES strategies in the domain help restrict the number of alternatives generated, but in many situations an alternative needs to be generated before it is tested for acceptance. Therefore a provision to generate all relevant alternatives in a KBES is an issue with scope for further work.

7. 2. 4 Other Paradigms of Computation

Other paradigms of computation, such as distributed processing systems could be explored to gain benefits for integrated computer-based design. Interaction of design teams could be used to model such distributed processing systems. Multiple views of a design artifact is an area that is suitable for multi-disciplinary design contexts. Further work on the view generation, with focus on a particular or selected discipline, would be a valuable extension of the current work. Agent-based systems, neural nets and genetic algorithms are new application areas in artificial intelligence. There is scope to investigate how such new computing paradigms can be of use to engineering and building design systems.

On the level of computer implementation, commercial database systems (RDBMS, OODBMS) could be considered as a basis for further work on integrated systems. DBMS by virtue of their wide use in corporate applications (including many in engineering), capacity to handle large volumes of data, ability to interact through graphical interfaces and client servers, seem ideal for practical large-scale integration in design or engineering enterprises.

CAD-based Knowledge-based System: After the formulation of the knowledge-base, Tall-D system implementation is done using a KBS development tool. It is possible however to develop Tall-D like systems with embedded knowledge-base modules in a traditional CAD environments such as Microstation or AutoCaD. Third party KBS tools integrated with such CAD packages can take advantage of the strong graphical engines of the latter for three-dimensional display and visualisation of alternative designs. Such an approach has the potential of making systems like Tall-D readily acceptable to users in the design and construction industry. In the current implementation of Tall-D the graphical functions slowed down the performance significantly even though two-dimensional representation is used.

There is a scope for collating, in a systematic way, the designs that are evaluated by experienced users of Tall-D and using it to refine the knowledge-base of Tall-D. Some of the alternatives that are always rejected by designers may not be generated to begin with. Also such a group of vetted designs can form a valuable repository for a case-based design system or evolve Tall-D as a dual paradigm system by incorporating case-based reasoning. Enhanced data processing and data management can increase KBES' design utility, especially if targeted for detailed design tasks in tall buildings design and other engineering design areas.

The use of ISO-STEP standard protocols for building and architecture, replacing the object library in Tall-D will make for an interesting exercise in Integrated design. Many issues in this context are worth experimenting with. Some of the issues are:

- Suitability of ISO-STEP for preliminary design purposes such as simultaneous multiple alternatives representation, generation and evaluation;
- Abstraction of design alternatives as coarse grained descriptions versus detailed descriptions and
- Graphical properties of the building objects to display themselves in a context sensitive manner.

ISO-STEP can also form the basis for interfacing with other programs such as those for detailed design. The current DXF interface feature can be replaced with an ISO-STEP model data file for a complete transfer of structural model information.

References

Akin, O., Dave, B. and Pithavadian, S. (1988). "Heuristic generation of layouts (HeGeL): Based on a paradigm for problem structuring", in J. S. Gero (ed), *Artificial Intelligence in Engineering: Design*, 413-444, Elsevier, New York, NY.

Allen, E. and Iano, J. (1989). "*The Architect's Studio Companion: Technical Guidelines for Preliminary Design*", John-Wiley and Sons, Inc., Toronto. p.468.

American Concrete Institute (1982). "*Lateral Load response of Buildings*," Report ACI-442R-71, ACI, Detroit, MI.

Anthony, W. R. (1985). "Concrete Buildings - New Formwork Perspectives," in J. Moreno (ed.) "*Analysis and Design of High-Rise Concrete Buildings*," SP-97, American Concrete Institute, Detroit, Michigan.

Augenbroe, G. (1994). "Integrated Building Design Systems in the Context of Product Data Technology," *Jnl. of Computing in Civil Engineering*, ASCE, 8(4), pp. 420-435.

Bailey, S.F. and Smith, F.C. (1994). "Case-Based Preliminary Design," *Jnl. of Computing in Civil Engineering*, ASCE, 8(4), pp. 454-468.

Barton, P. K. (1983). "*Building Services Integration*", E & F. N. Spon., New York, p.133.

Bédard, C. and Ravi, M. (1989). "Preliminary Planning and Design of Multistorey Buildings," *Computer-Aided Optimum Design of Structures: Applications*, Proc. of first International Conference, Southampton, UK, June 1989, pp.201-210.

Bédard, C. and Ravi, M. (1991). "Knowledge-Based Approach to Structural System Configuration for Multistorey Buildings," proc. *The Fourth International Conference on Computing in Civil Engineering*, July 29-31, Tokyo, Japan.

Bédard, C. and Ravi, M. (1992). "A Computer-Based Model for Integration in Building Design," proc. World Buildings Congress, Montreal, Canada.

Bennett, D.(1991). "Structural Developments: High Speed Concrete" *The Architects Journal*", 8 May, 1991, pp. 55-59.

Björk, B.C.(1989). "Basic Structure of a Proposed Building Product Model" *Computer Aided Design*, Vol.21, No.2, pp.71-78.

Björk, B.C.(1994). "RATAS Project - Developing an Infrastructure for Computer-Integrated Construction," *Journal of Computing in Civil Engineering*, ASCE, 8(4), pp. 401-419.

Building Owners and Managers Association of Québec - BOMA (1992). "*Montreal Office Space Directory 1992*," BOMA, Montreal, Quebec.

Building Systems Automation-Integration - BSAI (1991). Proceedings of First International Symposium, 2-8 June, University of Wisconsin - Madison, USA.

Cagan and Agogino (1989). "Why AI-design researchers should distinguish between creative, innovative and routine levels of design," Proc. of Workshop on Research Directions for Artificial Intelligence in Design, Stanford University, CA, USA.

Canadian Institute of Steel Construction - CISC (1995). "*Handbook of Steel Construction*," ed. M. I. Gilmore, CISC, 6th. Ed. Willowdale, Ontario.

Canadian Portland Cement Association (1995). "*Concrete Design Handbook*," Second edition, CPCA, Ottawa, Ontario.

Canadian Standards Association - CSA (1994). "Code of Practice for Concrete Structures A23.3-94," Canadian Standards Association, Ottawa, Onatrio.

Cervera, E. P. I. (1993) "KBS Validation: From Tools to Methodology," *IEEE Expert*, June 1993, pp. 45-47.

Cheong-Siat-Moy, F. (1991). "Analysis and Design of Structures with Outrigger Arms" in F. F. Monasa (ed.) *Approximate Methods and Verification Procedures of Structural Analysis and Design*, Amer. Soc. of Civil Engineers, New York.

Colaco, J. P. (1986)."Structural Systems Selections in High-Rise Buildings". in Lynn S. Beedle (ed.) *Advances in Tall Buildings*.

Construction Industry Institute (1987). "*Input Variables Impacting Design Effectiveness*," Design Task Force Report #8, Bureau of Engineering Research, Univ. of Texas, Austin, Texas.

COMBINE (1997)."What is COMBINE 2 about," World Wide Web document at URL: http://erg.ucd.ie/combine/what_about.html

Coull, A. and Stafford Smith, B. (1991). "*Tall Building Structures: Analysis and Design*", John Wiley, Toronto, pp. 537.

Council on Tall Buildings (1978-81). "*Planning and Design of Tall Buildings*", A Monograph, 5 Vols. ASCE, New York.

Council on Tall Buildings (1979). "*Structural Design of Tall Steel Buildings*", V. SB p. 38, ASCE, New York.

Council on Tall Buildings (1980). "*Structural Design of Tall Concrete and Masonry Buildings*", V. CB p.55, ASCE.

DMAC (1997). Design and Modelling Applications Council, World Wide Web URL: http://www.dmac.org.

Duchesne, D. P. J. and Humar, J. L. (1992). "Computer Integration in Civil Engineering", *Canadian Civil Engineer*, Oct., 1992, pp.3-4.

EDI (1997). "Electronic Data Interchange," World Wide Web URL: http://www.ediwi.ca/

Edlinger, P. "Importance of Data Exchange Standards for Process Industry," *Jnl. of Computing in Civil Engineering*, ASCE, 11(1), pp. 6-7.

Egan, M. D. (1975). "Concepts in Thermal Comfort," Prentice-Hall, New Jersey, p.203.

Eligator, M.H., Chu, S.L. and Occhicone, L. A. (1990). "Tall Buildings", Chapter 23, *Structural Engineering Handbook*, Gaylord, E.H. and Gaylord, C.N. (eds.), p.17-20.

Enseleit, J., Bitzer, D. and Pahl, P.J. (1995). "Experiences in using STEP for disstributed structural analysis", proc. *The Sixth international conference on Computing in Civil and Building Engineering*, Germany, 12-15 July. pp.337-342.

Evt, S. K., Khayyal, S. and Sanvido, V. E.(1992). "Representing Building Product Information Using Hypermedia," *Journal of Computing in Civil Engineering*, ASCE, 6(1), pp. 3-18.

Expert System (1997). URL: http://www.cs.cmu.edu/Web/Groups/AI/html/faqs/ai/expert/part1/faq.html

Fazio, P., Bédard, C. and Gowri, K. (1989) "Knowledge-Based System Approach to Building Envelope Design," *Computer-Aided Design*, 21(8), 519-27.

Fazio, P., Bédard, C. and Gowri, K. (1988) "Knowledge-based development tools for Processing Design Specifications," *Microcomputers in Civil Engineering*, 3(4), 333-344.

Fenves, S. J., Flemming, U., Hendrickson, C., Maher, M. L. and Schmitt, G. (1990). "Integrated Software Environment for Building Design and Construction", *Computer-Aided Design*, 22(1), 27-36.

Fenves, S. J. Rivard, H. and Gomez, N. (1995). "Conceptual Structural Design in SEED," *Journal of Architectural Engineering*, ASCE, 1(4), pp. 179-186.

Flemming, U., Coyne, R., Glavin, T. and Rychener, M. (1988). "A generative expert system for the design of building layouts - version-2", in J.S.Gero (ed.), *Artificial Intelligence in Engineering: Design*, 445-464, Elsevier, New York, NY.

Fling, R. S. (1987). "Preliminary Design," in *Practical Design of Reinforced Concrete*, pp. 457, John Wiley, NY.

Froese, T. (1996). "Models of Construction Process Information," *Journal of Computing in Civil Engineering*, ASCE, 10(3), pp. 183-193.

Gero, J. S., ed., (1988). "*Artificial Intelligence in Engineering: Design*," Proceedings of Int. Conf., Palo-Alto, CA, Aug., 1988, Elsevier, New York, NY.

Glover M. (1991). "Structural Developments: Steel or Concrete'" Architects Journal, 22 May 1991 pp. 47-51.

GoldHill (1989). "Goldworks II User Guide," GoldHill Computers, Inc., Chestnut Hill, Massachussettes.

GoldHill (1997). GoldHill World Wide Web Home Page http://www.goldhill-inc.com

Gowri, K. (1990). "Knowledge-Based Approach to Building Envelope Design," Ph. D. Thesis, Centre for Building Studies, Concordia University, Montreal, p.134.

Guise, D. (1990). "*Design and Technology in Architecture*", 2 ed., John Wiley & Sons, Inc., New York, NY.

Gupta, U. G. (1993). "Validation and Verification of Knowledge-Based Systems: A Survey", *Journal of Applied Intelligence*, 3(4):343-363.

Haber, D. and Karshenas, S. (1990)." CONCEPTUAL: An expert system for conceptual structural design," *Microcomputers in Civil Engineering*, 5(2):119-127.

Hanscomb (1995) "*Yardsticks for Costing*" Southam Building Reports, Toronto, Ontario.

Harty, N. and Danaher, M. (1997). "Evaluating preliminary structural designs in an expert system," *Computers & Structures*, 63(6) pp. 1243-1249.

Harmon, P., Maus, R. and Morrisey, W. (1988). "*Expert Systems Tools and Applications*," Addison-Wesley, New York, p.289.

Hauser, M. and Scherer, R.J. (1997). "Application of intelligent CAD paradigms to preliminary structural design," *Artificial Intelligence In Engineering* 11(3), pp. 217-229.

Hayes-Roth, F., Waterman, D. A. and Lenat, D. B., eds. (1983). "*Building Expert Systems*," Addison-Wesley, Reading, Massachussetts, p.444.

Herold, K. (1997). "Universal Building Language," *Jnl. of Computing in Civil Engineering*, ASCE, 11(1), pp. 1-3.

Holgate, A. (1986). "*The Art in Structural Design: An Introduction and Source Book*", Clarendon Press, Oxford, UK, p.337.

Intergraph (1997). "Intergraph: STEPping Up to Data Exchange", World Wide Web document at URL: http://www.intergraph.com/iss/products/pds/step, Intergraph Corporation.

ISO(1994). *STEP Part 1:Overview and fundamental principles*, ISO Standard 10303-1, Geneva, Switzerland.

Ito, K. (1995). "General Product Model and Domain Specific Product Model in the A/E/C Industry," in Mohsen, J.P.(ed.), "*Computing in Civil Engineering*" Proc. of Second Congress held in conjunction with A/E/C Systems '95, Atlanta, Amer. Soc. in Civil Engineers, New York, pp.13-16

Iyengar, H. S. (1977). "Preliminary Design and Optimization of Steel Building Systems," Technical Committee No. 14, ASCE/IABSE International Conference on Planning and Design of tall Buildings, Aug. 21-26, Bethlehem, PA, p.157

Iyengar, H. S. (1979). "Elastic Analysis and Design: Preliminary Design and Optimization", in *Structural Design of Tall Steel Buildings*, Beedle, L.S. (ed.) Vol. SB p.44, Council on Tall Buildings and ASCE, New York.

Jain, D., Krawinkler, H. and Law, K. H. (1993) "*Logic-based Conceptual Structural Design of Steel Office Buildings,*" Technical Report # 49, CIFE- Centre for Integrated Facility Engineering, Stanford University, Stanford, California.

Jayachandran, P. and Tsapastaris, N. (1988) "An Expert-system for the Selection of Tall Building Structural Systems", *Micro-computer Knowledge-Based Expert Systems,*" Proc. of Symposium on Expert Systems, H. Adeli (ed.), ASCE National Convention, Nashville, Tennesse.

Khan, F. R. and Iyengar H. S. (1973). "Optimization Approach for Concrete High-Rise Buildings", *Response of Multistorey Concrete Structures to Lateral Forces*, ACI Special Publication 36, 61-74.

Kantrowitz, M. (1997). "Answers to Frequently Asked Questions about Expert System Shells", comp.ai.shells, Feb. 1997, ftp.cs.cmu.edu:/user/ai/pubs/faqs/expert/expert_?.faq, mkant+ai-faq@cs.cmu.edu.

Korf R. E. (1987). "Planning as Search: A Quantitative Approach", *Artificial Intelligence*, Vol. 33, 65-88.

Kowalczyk, R. M., Sinn, R. and Kilmister, M. B. (1995). *Structural Systems for Tall Buildings*, Council on Tall Buildings and Urban Habitat, Pub. McGraw-Hill, p.422.

Kumar, V., and Topping B. H. V., (1988) INDEX - An Expert System for the Design of Industrial Buildings, "*Micro-computer Knowledge-Based Expert Systems*," Proc. of Symposium on Expert Systems, H. Adeli (ed.), ASCE National Convention, Nashville, Tennesse.

Lansdowne D. K. & Partners (1990) " *Lansdowne's Construction Cost Handbook*," Lansdowne & Partners, Toronto.

Liggett, R. S. (1990). "A designer Automated Algorithm Partnership An Interactive Graphic Approach of Facility Layout", *Proc. of the Third International Symposium on CAD*, March 23-29, Buffalo, New York, NY.

Lin, T. Y. and Stotesbury, S. D. (1988). "*Structural Concepts and Systems for Architects and Engineers*," 2 ed., John Wiley, p.507.

Means (1996) "Means Construction Cost Index," R. S. Means & Co, Kingston, MA.

Macleod, I.A. (1990). "Checking Models" in *Analytical Modelling of Structural Systems*, Ellir Horwood, England, pp. 124-127.

Maher, M. L., ed., (1987). "*Expert Systems for Civil Engineers: Technology and Applications*," American Society of Civil Engineers, New York, NY, p.148.

Maher, M. L. and Fenves, S. J. (1984). "HI-RISE: A Knowledge- Based Expert System For The Preliminary Structural Design of High Rise Buildings," *Report No.R-85-146*, Dept. of Civil Engineering, Carnegie Mellon University, Pittsburg, PA.

Maher, M. L. and Garza, A.G. (1996). "Developing Case-based Reasoning for Structural Design," *IEEE Expert*, Vol.11, No.3, pp. 42-52.

Menniti, J.P. (1996). "Using Computers Effectively in today's Civil Engineering Office," *Jnl. of Computing in Civil Engineering*, ASCE, 10(4), pp. 261-262.

Mitchell, W.J. (1977). "Automated Spatial Synthesis" in *Computer-aided Architectural Design*, Van-Nostrand, New York, NY, p.573.

Monasa, F. F., ed., (1991). *Approximate Methods and Verification Procedures of Structural Analysis and Design*, Amer. Soc. of Civil Engineers, New York, p.91.

Moreno, J. (1985). "*Analysis and Design of High-Rise Concrete Buildings*," Special Publication 97, American Concrete Institute, Detroit, Michigan, p.330.

NBCC (1995), *National Building Code of Canada*, National Research Council of Canada, Ottawa, p.497.

Neil, J. M. (1982). "Approximate Estimating," in *Construction Cost Estimation for Project Control*, Prentice-Hall, Englewood Cliffs, New Jersey, p.331.

Nilson, A. H. and Winter, G. (1986). "Design of Concrete Structures," Tenth edition, McGraw Hill, NY, p.730.

Oxman, R. E. and Oxman, R. M. (1992). " Refinement and Adaptation in Design Cognition", Design Studies, Vol. 13, No. 2, pp. 117-134.

Paek, Y. J. and Adeli, H. (1988). "SDL: An Environment for Building Integrated Structural Design Expert Systems." *Micro-computer Knowledge-Based Expert Systems*," H. Adeli (ed.), ASCE, New York.

PDES (1997). "Product Data Exchange Using STEP," World Wide Web URL: http://www.scra.org/pdesinc.html

Poon,J and Maher, M.L.(1997). "Co-evolution and Emergence in Design," *Artificial Intelligence In Engineering* 11(3), pp.319-327

Portland Cement Association - PCA (1992). "*Simplified Design: Reinforced Buildings of Moderate Size and Height*," Portland Cement Association, Skokie, Illinois, p.256.

Radford, N. and Gero, J. S. (1988). "*Design by Optimization in Architecture, Building and Construction*," Van Nostrand, NY.

Ravi, M., Didsbury, R., Sharma, S.N., and Romaniszyn, T. (1997). "Design of Nuclear Class Piping Supports in a 3D CADDS Environment", working paper, Engineering Tools, Atomic Energy Canada Ltd., Chalk River Labs, Chalk River, Ontario.

Robertson L. E. and See S. T. (1987). "Preliminary Design of High-Rise Buildings," in R. N. White and C. G. Solomon, eds., *Building Structural Design Handbook*, John Wiley.

Rush, R. (1986). "*The Building Systems Integration Handbook*", American Institute of Architects, John Wiley & Sons, New York, p.445.

Rychener M. D. (1988) (ed.) "*Expert Systems for Engineering Design,*" Academic Press, New York, p.306.

Sause, Richard. (1989). "A Model of the Design Process for Computer Integrated Structural Engineering", *Ph. D. Thesis,* Civil Engineering, Univ. of California, Berkeley, p. 232.

Schollar, T. (1989). "Structural sizing: Rules of thumb,"*The Architects Journal*, 6 Spet. 1989, pp. 69-71.

Schueller, W. (1986). "*High-Rise Building Structures*", 2nd. ed., R. E. Krieger, Malabar, Florida, p.274.

Sommerville, I. (1993) "Validation and Verification" Part-4 in *Software Engineering,* 4th ed., Addison-Wesley, Don Mills.

Spires, D. and Arora, J. S. (1990). "Optimal Design of Tall RC-Framed Tube Buildings," *Journal of Structural Engineering.,* 116(4), pp.877-97, ASCE.

Sriram, D. (1986). *Knowledge-Based Approaches for Structural Design*, Ph. D. Thesis, Dept. of Civil Engineering, Carnegie-Mellon University, Pittsburg, PA.

Sriram, D., Logcher, R. D., Wong, A. and Ahmed, A. (1990). "A Case study in Computer-Aided Cooperative Product Development," *Tech. Report No: IESL-90-01,* Intelligent Engineering Systems Lab., Dept. of Civil Engrg., MIT, Boston MA.

Steyert, R. (1972). "*The Economics of High-Rise Apartment Buildings of Alternate Design Configurations*," Amer. Soc. of Civil Engineers, New York, NY.

Supplement to NBCC (1995) - *National Building Code of Canada*, National Research Council of Canada, Ottawa.

Taranath, B. S. (1988). "*Structural Analysis and Design of Tall Buildings*", McGraw Hill, p.739.

Timberline (1997). *Presision Estimating Software - Standard Edition*, Timberline Software Corporation, Beaverton, Oregon, USA.

Trebilcock, P., Lawson, M and Owens, G. (1991a). "Structural Developments: Steel Frame Design" *The Architects Journal*", 24 April 1991, pp. 47-53.

Trebilcock, P., Lawson, M. and Owens, G. (1991b). "Structural Developments: Steel Frame Buildings" *The Architects Journal*", 1 May 1991, pp. 55-56.

Turkiyyah, G.M. and Fenves, S.J. (1996). "Knowledge-based Assistance for Finite-Element Modeling," *IEEE Expert*, Vol.11, No.3, pp. 23-32.

Turk, Z., Isakovic, T. and Fischinger, M. (1994). "Object-oriented Modeling of Design System for RC Buildings," *Jnl. of Computing in Civil Engineering*, ASCE, 8(4), pp. 436-453.

White, R. and Salmon, C. G., eds. (1987). "Vertical Transportation - Structural Aspects", in *Building Structural Design HandBook*, John Wiley & Sons, Inc., New York, NY.

Willey, D.S. (1978). "Structure for automated sketch design", *Computer-Aided Design*, 10(5), 307-312.

Wygant, D.F. (1996). "OLE for design and modeling," *Windows NT Magazine*, March 1996. Also available as World Wide Web document at http://www.winntmag.com/issues/Mar96/OLE.html.

Zualkernan, I. A. and Lin, Y. J. (1993). "Experimental Evaluation of Output Based partition Testing for Expert Systems", in Proc. of the 1993 IEEE International Conf. on Tools for AI, pp. 190-197, Boston, Massachusetts, Nov. 1993.

Appendix A: Knowledge-Based Systems

Knowledge-based systems (KBS) can be used as computer-based design aids. They are able to incorporate expertise and information derived from experience as well as knowledge of the state-of-the-art in a given domain. Thus they provide expert assistance to designers in the domain. Research and development in such Knowledge-Based Design applications in engineering that are of recent origin holds a good prospect for solving many design automation problems. In this chapter a brief review of the different elements of a KBS and variations in their architecture as well as a review of related work in KBS is presented.

A-1 Components of a Knowledge-Based System

The principal components of a typical knowledge-based system are shown in Figure A.1. Different KBS development tools provide facilities to implement the components of the KBS. Following is a description of the main components of a complete KBS.

The knowledge-base contains facts of the domain in declarative form. Current knowledge-base systems have standardised on the hybrid knowledge representation scheme - frames and production rules. A template for facts/values are frames (or alternatively called objects/schemas). Heuristics are described in the form of IF-THEN rules. Predefined values in the knowledge-base may also be specified in the frames. The context is the component of the system that holds the specific information, regarding the current problem being examined. The inference engine (or inference mechanism) is the component that contains the control information. It uses the information in the knowledge-base to expand and upgrade the context. Forward chaining, backward chaining and a

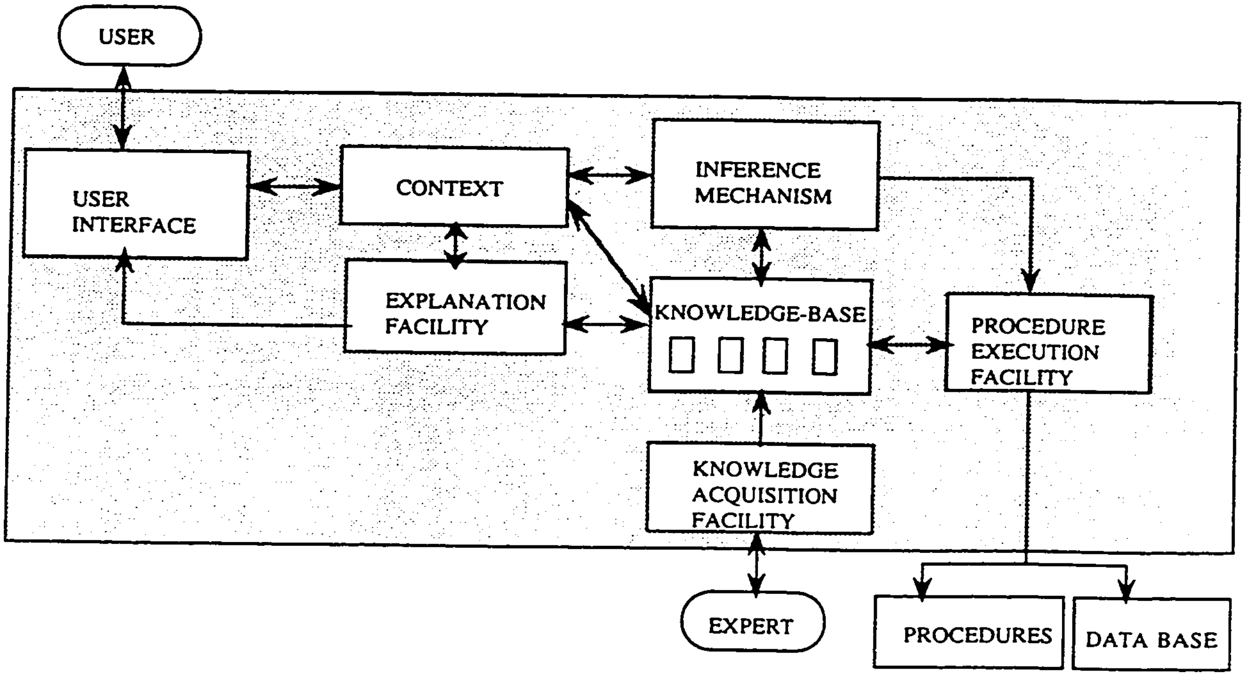


Figure A-1 Principal Components of a Typical Knowledge-Based System.

combination of the two are the possible ways in which the rules may be used in the reasoning process by the inference engine. The explanation facility enables the system to explain why a piece of information is needed or how a decision or solution was arrived at. The knowledge acquisition facility is the component that incorporates the user-supplied rules and declarative parts into the knowledge-base. This can take the form of a simple editor to an intelligent interface to help the expert define the knowledge-base. The user interface facilitates the definition of context, provides a transparent picture of the functioning of the system, either in the form of text or graphical display related to the problem and finally gives the solution or a summary of it. An overview of the system used in the current research for integrated building design is presented in Chapter 3.

A-2 Problem Solving in KBS

There are different problem solving techniques that can be followed in developing a KBS. Depending on the nature of the problem and solution needed, there may be an approach more appropriate to adopt than others. Maher (1987) gives an overview of the different approaches and control strategies.

The two broad categories of approach are derivation and formation methods. Derivation approach consists of selecting an appropriate solution from among a group of possible ones. With the available data, an inference network can be used to find the solution. Diagnosis and classification are examples of this approach. Ways of implementing this approach are forward chaining, backward chaining and mixed chaining.

Formation approach involves the generation of a solution using the available facts in the knowledge-base upon initiation. In this method the solution is generated after the input data is supplied, rather than having a pool of predefined solutions as in the previous approach. Synthesizing submodules to form a complete solution is required in the formation approach. Planning as well as design systems mostly employ this approach. Ways of implementing this approach is by generate-test, problem reduction and agenda control. In the current design system, the formation approach is implemented by means of hierarchical-generate-test.

Control Strategies

There are a number of ways of implementing a control strategy in knowledge-based systems. These are forward chaining, backward chaining, mixed chaining, problem reduction, generate-test and agenda control.

Forward chaining involves the use of the initial state of the knowledge-base to assert more facts into the knowledge-base and to modify the knowledge-base. The final state of the knowledge-base will contain the solution to the problem. This chaining is also called data-driven. The present system utilises forward chaining within each group of rules. Each group of rules focuses on a particular aspect of building design.

Backward chaining is a method in which a goal (or hypothesis) is set as true and a search of the knowledge-base is conducted to ascertain its feasibility. It is also known as goal-driven. Intermediate sub-goals may be set up to validate further unknown facts. Satisfying the requirements of all the sub-goals in the system by the known facts in knowledge-base leads to the initial hypothesis being established as true. An alternative hypothesis is set as true in case of a failure in the previous attempt.

Mixed chaining, a strategy where both the above approaches are combined can also be used. Using the available data, forward chaining will establish the problem solving strategy. The result is a set of possible goal states that the system could be attempt to establish. The sequence of goals pursued is preferably established by the forward chaining operation. Then the backward chaining seeks to establish a goal as feasible based on the available data. Sub-goals are set up in the process to establish any unknown data until the initial goal is established.

Problem reduction involves the decomposition of the problem into sub-problems characterised by an AND-OR network. Traversing the different branches of the network generates the solution. Branches at an AND node represent components that are part of the solution, while branches at an OR node represent alternatives. Traversing an AND node means seeing that the successive nodes return a value TRUE (in the case of OR node, at least one alternative successive node should return TRUE).

Generate-test is the technique where all possible solutions are exhaustively generated and tested until one satisfies the specification for the solution. To improve the efficiency of the system, modified versions of the paradigm such as plan-generate-test and hierarchical generate-test are developed. The former prescribes pruning at the outset to eliminate inconsistent solutions. The latter emphasises pruning on a continual basis, where in each step the solution becomes more detailed. Thus, pruning partial solutions on the way to the final solution(s) is efficient as it uses new information generated in the previous cycle. This particular approach is used in the implementation of the system, since it considers a wider range of initial solutions and yet generates only limited number of alternatives in the end.

Agenda control is a useful strategy in which priority is assigned to the tasks in the KBS. The task with the utmost priority is carried out first and the remaining tasks in order of current priority. This type of agenda control is useful in selecting between different tasks. It is especially useful in coordinating multiple sources of knowledge represented by many groups of rules in the knowledge base. In the implementation of the current system it is used to activate groups of rules arranged hierarchically.

A-3 Tools for KBS Development

Knowledge-based systems can be developed with a variety of tools that are available now (Harmon, Maus and Morrisey 1988). Starting with general purpose programming languages (eg. C, FORTRAN and LISP), general purpose representation languages (eg. PROLOG and OPS5), KBS building shells (eg. NEXPERT, LEVEL5, INSIGHT2, GURU etc.) and the more versatile KBS development environments (eg. GOLDWORKS-II, KNOWLEDGE-CRAFT, ART and KEE) which have various degrees of sophistication as development tools. Increasing sophistication of the tool entails more powerful hardware, but offers enhanced facilities to develop the various components of the system. The tool used in the current work is GOLDWORKS-II, a LISP-based development environment, compares very well with any other tool, particularly on a personal computer (Fazio, Bédard and Gowri 1989). It offers such features as frames and production rules to build the knowledge-base, a graphical interface for knowledge acquisition, operations that could be used to access specific values in the knowledge-base, and finally to enable development of design interface and end-user interface. GOLDWORKS-II is one of the most versatile KBS tools available on advanced personal computers, offering a powerful development environment.

A list of home pages of major commercial KBES tools vendors is maintained as a World Wide Web hyper text document (Expert System 1997). This document can also be obtained by email or ftp (Kantrowitz 1997).

A-4 Glossary of Knowledge-based System Terms

Agenda control: Use of higher level rules incorporating knowledge about the design process to enable the inference engine to prioritize tasks during program execution.

Antecedent: The IF part of a rule.

Attribute: A data member of a frame, representing a quality of an artifact by assigning numeric or textual value.

Axiom: A basic truth or a given fact in the domain of the KBS.

Backward Chaining: The process of matching the THEN part of the rule with facts (assertions and axioms) in the knowledge-base and if successful, asserting the IF part as a new assertion and continue the process until all rules with matching THEN part have their IF part entered as assertions.

Consequent: The THEN part of a rule.

Constraint: Limit on the value of an attribute of an object. It can be a single value, a range or a list of values.

Context: The combination of axioms, dynamically generated objects and their attribute values (representing among other, user/designer constraints and input) in the knowledge-base. In a design KBS this becomes the design context.

Demon function: A (Lisp) function bound to an attribute of an object that is triggered upon an event related to the attribute value.

Domain: The area of specialization the KBS is deployed in.

Forward Chaining: The process of matching the IF part of the rule with the facts (assertions and axioms) in the knowledge-base and executing the THEN part of the rule and continuing to do the same until all matching rules are fired.

Frame: It is the conceptual data entity that encapsulates both data and the related procedures. It is the name used for *object* in the Common Lisp Object System (CLOS). Frame/Object has data members (attributes), member functions and typical features of object-oriented systems such as inheritance, polymorphism etc. GoldWorks is based on CLOS.

Generate-test: A problem solving method widely used in computer-based design and planning assistants. Generation module of the KBS produces design proposal(s). The Test module evaluates based on a set of criteria, resulting in a rating or accept/reject decision of the design proposal. This method when used on different abstractions of a design problem, is known as hierarchical generate-test.

Handler function: Also known as member function, it is part of the Frame. It incorporates procedural knowledge related to the artifact modeled by the Frame. It can only be invoked by sending a message to the instance of the Frame during run-time.

Hybrid knowledge representation: Use of frames and rules to declare the domain knowledge-base, creating a more efficient knowledge representation compared to rule-based systems.

Inheritance: When different frames are related in a hierarchy, frames in the tree acquire the properties (attributes, member functions etc.) of all parent frames as well as similarly acquired properties of those parent frames. Attribute constraints can be controlled so as to merge with or override any locally defined values.

Inference engine: The component of the knowledge-based system that uses IF-THEN rules to infer and generate new facts into the knowledge-base until all rules with matching antecedents are fired.

Instance: Run-time copy of a frame with attribute values (slot values). Attribute values can only be assigned using instances.

Knowledge-base: The rules, frames, dynamic instances of frames, assertions generated by rules and axioms predefined all form the contents of a knowledge-base representing knowledge in a domain.

Knowledge-based System (KBS): A computer-based system that uses formally declared knowledge in the form of rules and frames to solve a problem in a domain. KBS generally show a broad knowledge of a domain as opposed to the deep/narrow knowledge exhibited by expert systems. However the two terms are often used interchangeably.

Macro rule: A rule incorporating the overall solution strategy for a (design) problem. It is in a way a rule for the use of the rule-base.

Member function: Same as a handler function. See handler description of handler function.

Meta-rule: Same as a Macro rule, defined earlier.

Mixed Chaining: A combination of forward chaining and backward chaining, where sub-goals are setup as a backward chaining tasks and the rest of the rules function as data driven forward chaining rules.

Object: Same as Frame. See description of frame.

Polymorphism: A feature of object oriented programming that enables the same function name (eg. a member function) or variable name (eg. an attribute) to be used to implement versions of the same. The variable type, function parameter list and/or return type determine which variable is used or which function executes.

Problem Reduction: Dividing or decomposing a (design or planning) problem into smaller and smaller tasks where a solution to all the smallest tasks cascade into a solution for the whole upon recomposition of the results.

Rules: Formally defined IF-THEN statements describing the domain knowledge.

Rule-set: A formally defined collection of rules, that would all normally be used in conjunction and all pertain to a sub-problem of design.

Slot: Same as attribute. See description of *Attribute*.

Sponsors: Formal entity in GoldWorks that is used to partition the rule-base. By enabling and disabling a sponsor the rule-sets and/or rules under the sponsor are entered and withdrawn respectively from the inferencing process. This relieves the inference engine from processing irrelevant rules as the user proceeds with different tasks in the KBS.

***User interface*:** An interface for the human-computer interaction. The most common one is the graphical and windowed system or the Graphical User Interface (GUI). However it is not unique to KBS.

APPENDIX B Tall-D Design Example of Chapters 4 and 5*

TABLE OF CONTENTS: Appendix B

* Note: The first half of the output for this example is presented in Table 4.1 in Chapter 4. Hence it is not repeated here. Therefore only the structural design information follows. Sample images for Layout#25 are in thesis text, Chapter 5.

TABLE OF FIGURES: Appendix B

SUMMARY OF STRUCURAL LAYOUT ALTERNATIVES

(Note: Sample alternatives have also been presented graphically at the end)

Alternative Structural Scheme for: Layout#25

```
Alternative Structural Scheme for: Layout#25
Layout#25: Braced-Perimeter-5
 Structure Type: Braced Frame Scheme
 Major Stability Element: Braced-Frame-Perimeter
 Column LayoutType: Perimeter-Based
   Clear Span between core and window line:
       Front & Back edges to Core: 12.3m
   Clear Span between core and window line:
       Left & Right edges to Core: 12.3m
  1.Column Spacing:
     Perimeter (Front & Back):10.0m
     Perimeter (Sides):10.0m
  2.Column Spacing:
     Perimeter (Front & Back):5.7m
     Perimeter (Sides):5.7m
   Structural Grid inside the Core area: 5.2m x 5.2m
```

```
Alternative Structural Scheme for: Layout#25
Layout#25: Braced-Internal-6
 Structure Type: Braced Frame Scheme
 Major Stability Element: Braced-Frame-Internal
 Column LayoutType: Grid-2d
 Grid with unequal column spacing
   Alternative Grid Spacings:
   Bay: 5.7m  Aisle: 13.3m
 Column LayoutType: Grid-Aligned-To-Core
   Clear Span between core and window line:
       Front & Back edges to Core: 12.3m
   Clear Span between core and window line:
       Left & Right edges to Core: 12.3m
  1.Column Spacing:
     Perimeter (Front & Back):5.7m
     Perimeter (Sides):5.7m
   Structural Grid inside the Core area: 5.2m x 5.2m
```

```
Alternative Structural Scheme for: Layout#25
```
Layout#25: Steel-Scheme-8
```
Structure Type: Steel Rigid Frame Scheme
Major Stability Element: Perimeter-Frame
Column LayoutType: Perimeter-Based
  Clear Span between core and window line:
      Front & Back edges to Core: 12.3m
  Clear Span between core and window line:
      Left & Right edges to Core: 12.3m
 1.Column Spacing:
    Perimeter (Front & Back):3.1m
    Perimeter (Sides):3.1m
 2.Column Spacing:
    Perimeter (Front & Back):4.4m
    Perimeter (Sides):4.4m
  Structural Grid inside the Core area: 3.1m x 3.1m
```

Approximate Column Sizes for: Layout#25

Structural Scheme: Steel-Scheme-7

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 4.4m Aisles: 13.3m

STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
22-22	(W150 150)	(W150 150)	(W150 150)	(W150 150)
21-19	(W200 200)	(W200 200)	(W200 200)	(W200 200)
18-16	(W250 250)	(W250 250)	(W310 310)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)	(W200 200)
12-10	(W310 310)	(W310 310)	(W310 310)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Column Steel weight for this alternative: 206400.0 KG

Alternative with internal grid of Bays: 5.7m Aisles:13.3m

STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
22-22	(W150 150)	(W150 150)	(W150 150)	(W150 150)
21-19	(W200 200)	(W200 200)	(W250 250)	(W200 200)
18-16	(W250 250)	(W250 250)	(W310 310)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)	(W250 250)
12-10	(W310 310)	(W310 310)	(W360 360)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W460 280)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Column Steel weight for this alternative: 168000.0 KG

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

```
Alternative with Perimeter column spacings XX:4.4m YY:4.4m
 STEEL Strength: 220MPa
--------------------------------------------------
 Storeys   PerimeterXX   PerimeterYY    Corner
  in Group     (mm)          (mm)          (mm)
--------------------------------------------------
 22-22   (W150 150)   (W150 150)   (W150 150)
 21-19   (W200 200)   (W200 200)   (W200 200)
 18-16   (W200 200)   (W200 200)   (W200 200)
 15-13   (W310 310)   (W310 310)   (W310 310)
 12-10   (W310 310)   (W310 310)   (W310 310)
  9- 7   (W310 310)   (W310 310)   (W310 310)
  6- 4   (W310 310)   (W310 310)   (W310 310)
  3- 1   (W310 310)   (W310 310)   (W310 310)
--------------------------------------------------
 Column Steel weight for the above alternative: 196800.0 KG
```

```
Alternative with Perimeter column spacings XX:5.7m YY: 5.7m
 STEEL Strength: 220MPa
--------------------------------------------------
 Storeys   PerimeterXX   PerimeterYY    Corner
  in Group     (mm)          (mm)          (mm)
--------------------------------------------------
 22-22   (W150 150)   (W150 150)   (W150 150)
 21-19   (W200 200)   (W200 200)   (W200 200)
 18-16   (W250 250)   (W250 250)   (W250 250)
 15-13   (W310 310)   (W310 310)   (W310 310)
 12-10   (W310 310)   (W310 310)   (W310 310)
  9- 7   (W310 310)   (W310 310)   (W310 310)
  6- 4   (W310 310)   (W310 310)   (W310 310)
  3- 1   (W310 310)   (W310 310)   (W310 310)
--------------------------------------------------
 Column Steel weight for the above alternative: 168000.0 KG
```

Structural Scheme: Steel-Scheme-8

Column Layout Type: Perimeter-Based

Details of Columns :

Alternative with Perimeter column spacings XX:4.4m YY:4.4m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
22-22	(W150 150)	(W150 150)	(W150 150)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 196800.0 KG

Alternative with Perimeter column spacings XX:3.1m YY:3.1m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
22-22	(W150 150)	(W150 150)	(W150 150)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 254400.0 KG

Structural Scheme: Fr-Shearwl-Scheme-9

Column Layout Type: As-Per-Shearwalls

Shearwall Details:

```
 Number of Shearwalls (with flanges) in Core: 3
 Concrete strength: 40MPa
 Spacing of Shearwalls:7.7m
 1. Thickness of wall at base: 450mm
    Wall Shape: C
 2. Thickness of wall at base: 375mm
    Wall Shape: I
 3. Thickness of wall at base: 450mm
    Wall Shape: C-INVERTED
 Shearwall concrete volume for this alternative:  1606.
cu.m.
   All Walls may taper to 250mm at top;
   May be terminated in some cases;
   & also in cases where elevator banks drop off.
```

Details of Columns:

Alternative with Perimeter column spacings XX:4.4m YY:4.4m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
22-22	300	300	300
21-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	300	300	300
9- 7	400	400	400
6- 4	550	550	550
3- 1	550	550	550

Column and Shearwall concrete volume for this alternative:
2248. cu.m.

Alternative with Perimeter column spacings XX:6.7m YY:6.7m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
22-22	300	300	300
21-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	400	400	400
9- 7	550	550	550
6- 4	650	650	650
3- 1	700	700	700

Column and Shearwall concrete volume for this alternative:
2335. cu.m.

Alternative with Perimeter column spacings XX:5.7m YY:5.7m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
22-22	300	300	300
21-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	400	400	400
9- 7	550	550	550
6- 4	550	550	550
3- 1	650	650	650

Column and Shearwall concrete volume for this alternative:
2312. cu.m.

Structural Scheme: Braced-Internal-6

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays:5.7m Aisles:13.3m
STEEL Strength: 220Mpa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
22-22	(W150 150)	(W150 150)	(W150 150)	(W150 150)
21-19	(W200 200)	(W200 200)	(W250 250)	(W200 200)
18-16	(W250 250)	(W250 250)	(W310 310)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)	(W250 250)
12-10	(W310 310)	(W310 310)	(W360 360)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W460 280)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Column Steel weight for this alternative: 168000.0 KG

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX:5.7m YY:5.7m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
22-22	(W150 150)	(W150 150)	(W150 150)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W250 250)	(W250 250)	(W250 250)
15-13	(W310 310)	(W310 310)	(W310 310)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 168000.0 KG

Table B1: Summary of gravity system alternatives for Layout#25

Construction Material	Type of Gravity System	Floor Depth OR Beam/slab size mm x mm	Relative Prefereance	Column Layout Type
Concrete Floor Systems		Structural Scheme: Fr-Shearwl-Scheme-9		
	One-Way Beam-Slab	1000 deep 200 slab	3	As-per-Shearwalls
	Joist Slab	900deep 260 joists	1	As-per-Shearwalls
	Waffle Slab	300 slab	2	As-per-Shearwalls
	Band-Beam - Slab	820 band 170 slab	4	As-per-Shearwalls
	Hollow-Core Slab	320 slab	5	As-per-Shearwalls

Note: 1 is most preferred; Based on historical in-place cost, on a relative basis.

...continued

Table B1: Summary of gravity system alternatives for Layout#25 (continued).

		Structural Scheme: Steel-Scheme-7			Structural Scheme: Braced-Internal-6		
Composite Steel Deck Floor Systems with Steel Beams	Tapered Beam	250x510	3	Grid-2D	250x510	3	Grid-2D
	Truss Beam	610deep +60 deck	1	Grid-2D	610deep +60 deck	1	Grid-2D
	Haunch Beam	250x410	2	Grid-2D	250x410	2	Grid-2D
	Parallel Beam	200x410	4	Grid-2D	200x410	4	Grid-2D
	Stub Girder	250x510	5	Grid-2D	250x510	5	Grid-2D
	Tapered Beam	230x470	3	Grid-Aligned-To-Core	230x470	3	Grid-Aligned-To-Core
	Truss Beam	560 deep +60 deck	1	Grid-Aligned-To-Core	560 deep +60 deck	1	Grid-Aligned-To-Core
	Haunch Beam	230x380	2	Grid-Aligned-To-Core	230x380	2	Grid-Aligned-To-Core
	Parallel Beam	190x380	4	Grid-Aligned-To-Core	190x380	4	Grid-Aligned-To-Core
	Stub Girder	230x470	5	Grid-Aligned-To-Core	230x470	5	Grid-Aligned-To-Core
		Structural Scheme: Steel-Scheme-8					
	Tapered Beam	230x470	3	Perimeter-Based			
	Truss Beam	560 deep +60 deck	1	Perimeter-Based			
	Haunch Beam	230x380	2	Perimeter-Based			
	Parallel Beam	190x380	4	Perimeter-Based			
	Stub Girder	230x470	5	Perimeter-Based			

Sample Images of Alternative for Layout#14

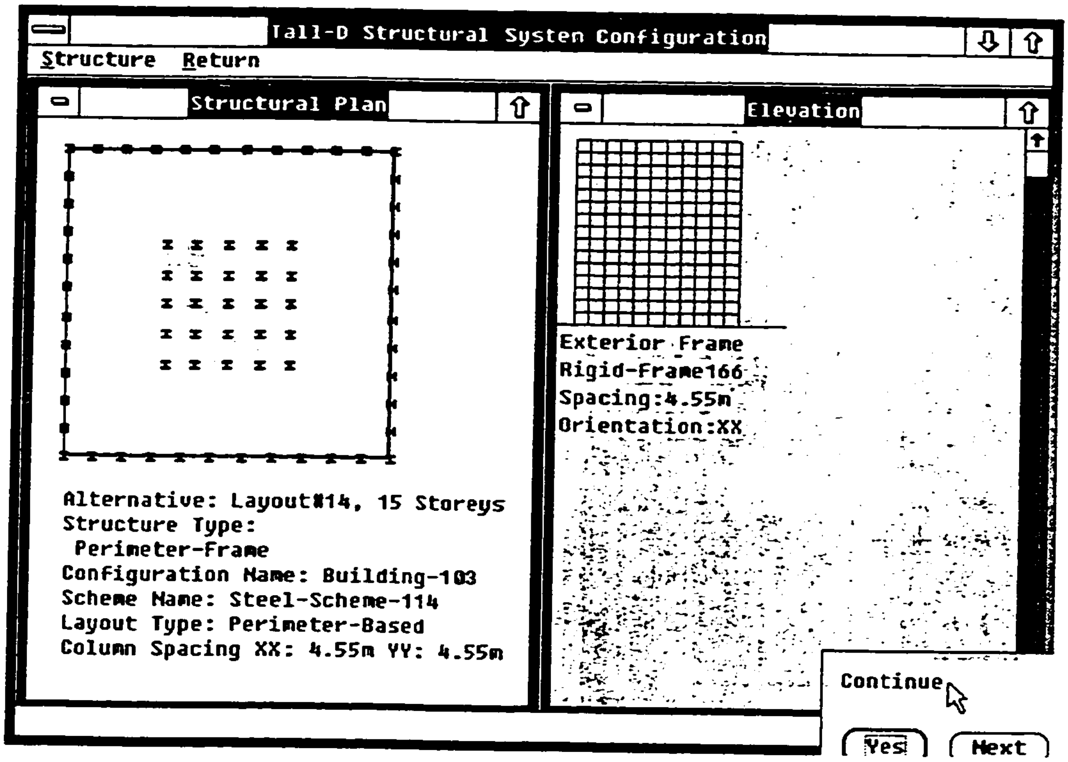


Fig. B- 1 Perimeter-based rigid frame strucural system alternative for Layout#14
Perimeter column spacing 4.55m and 4.55m

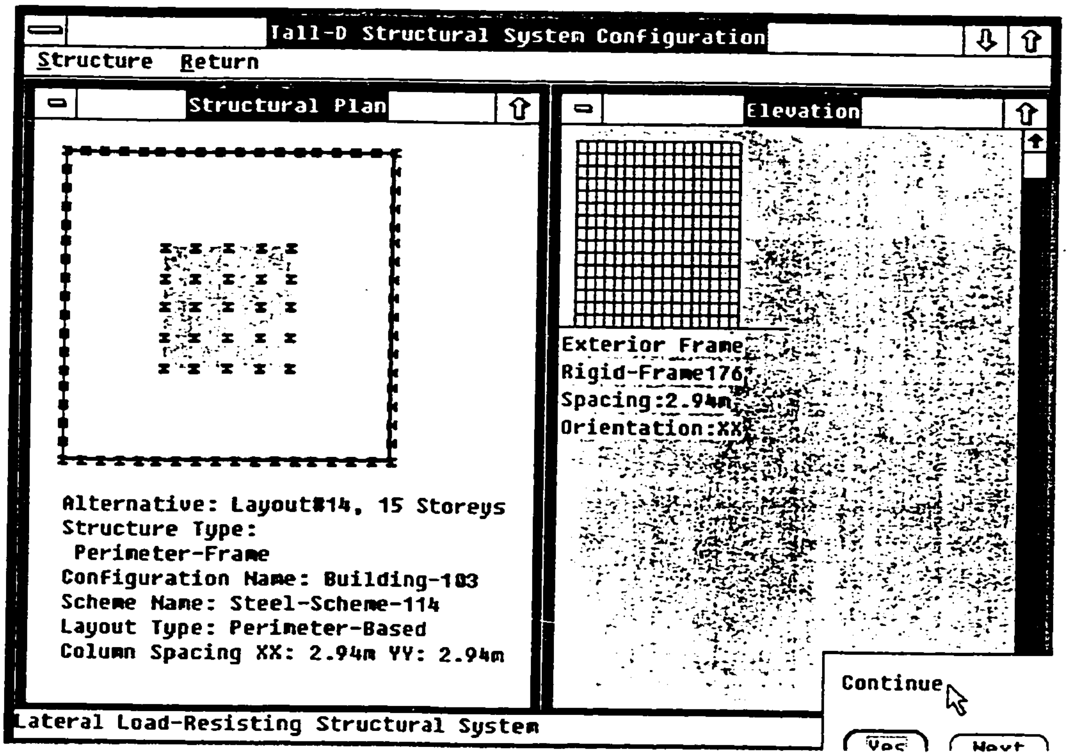


Fig. B- 2 Perimeter-based rigid frame strucural system alternative for Layout#14:
Perimeter column spacing 2.94m and 2.94m

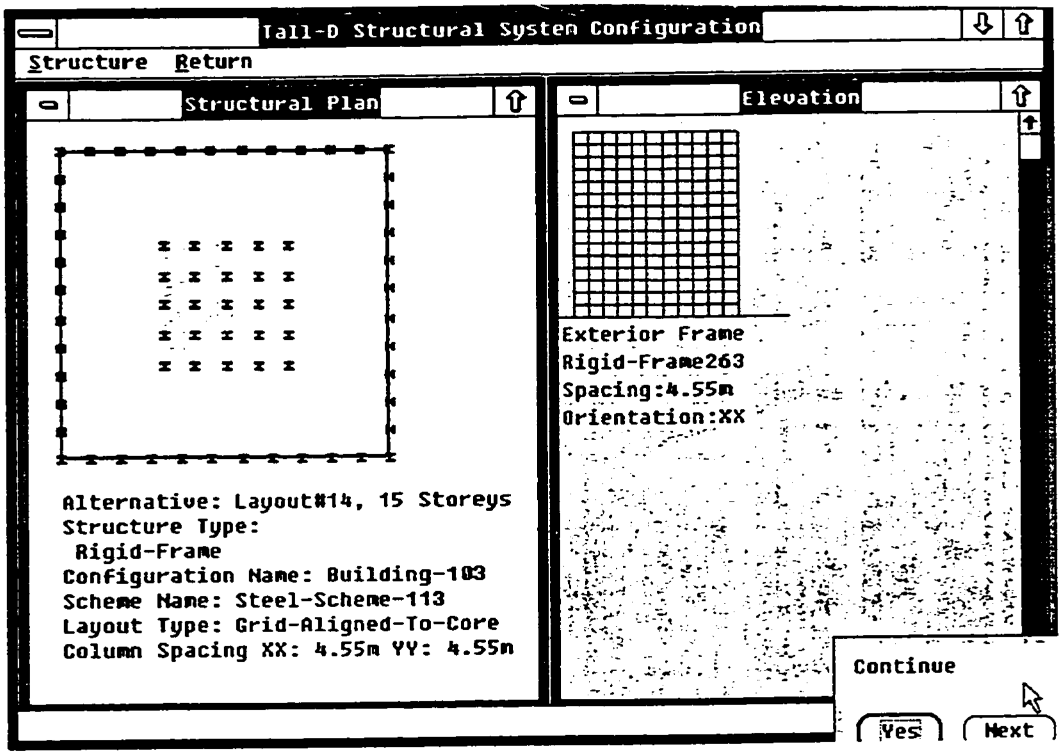


Fig. B- 3 Rigid frame strucural system alternative for Layout#14: Perimeter column spacing 4.55m and 4.55m

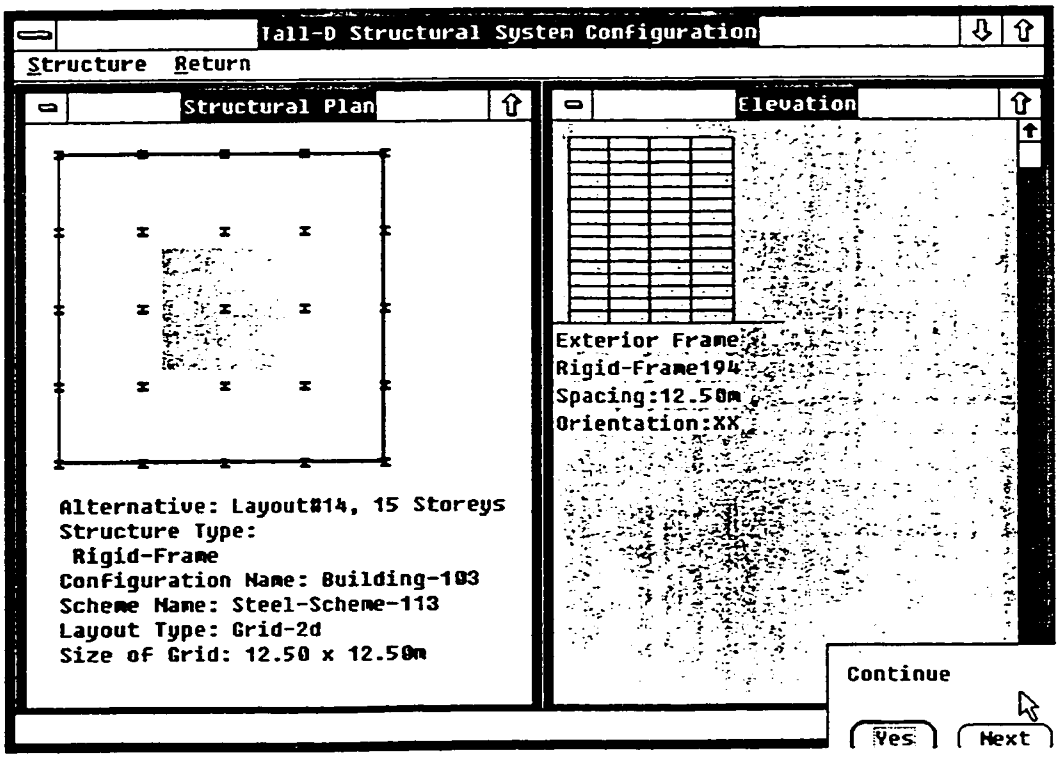


Fig. B- 4 Rigid frame strucural system alternative for Layout#14: Grid 12.5mx12.5m.

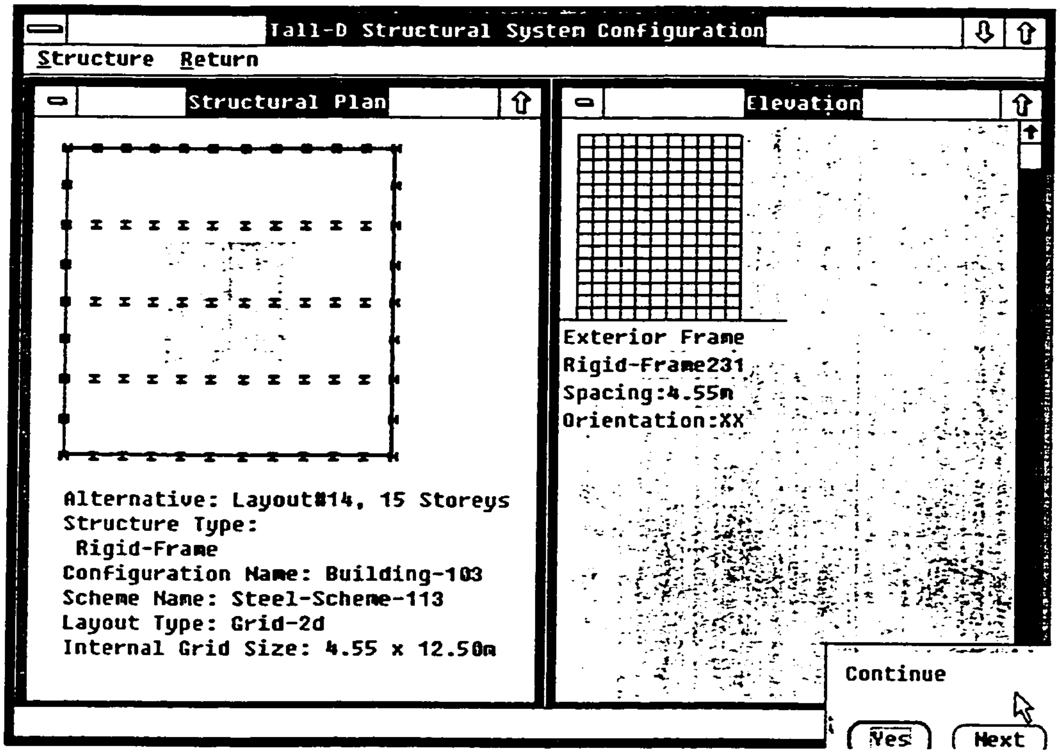


Fig. B- 5 Rigid frame strucural system alternative for Layout#14: Grid 4.55mx12.50m

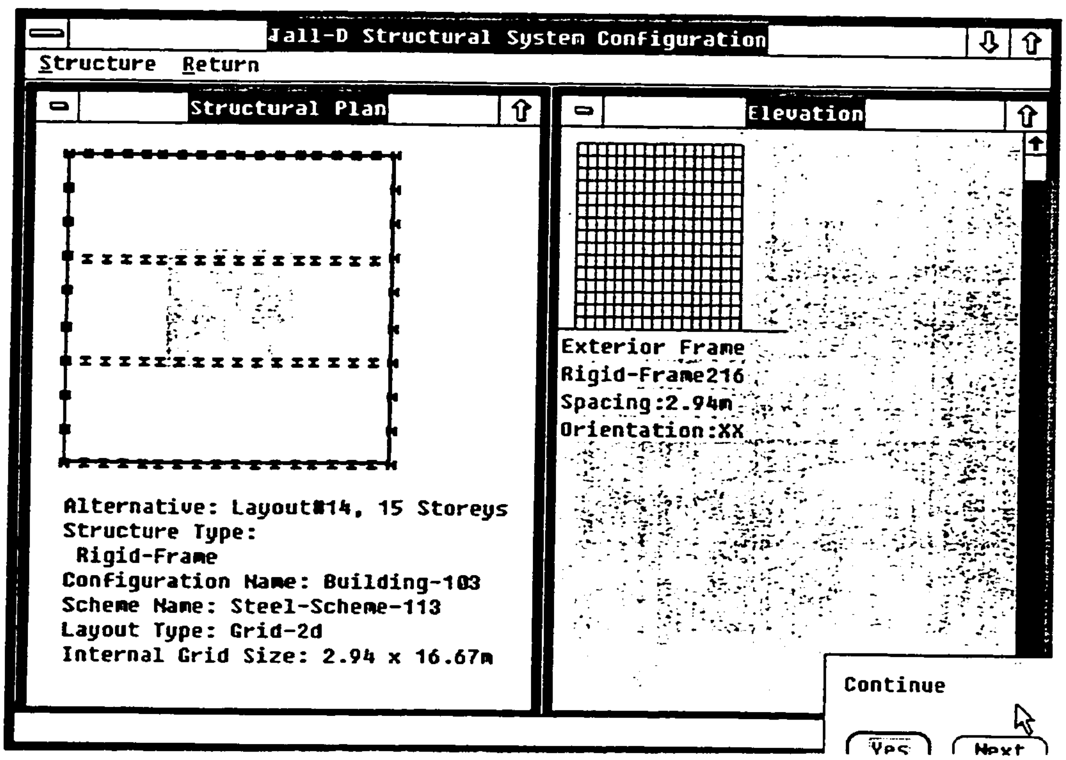


Fig. B- 6 Rigid frame strucural system alternative for Layout#14: Grid 2.94mx16.67m

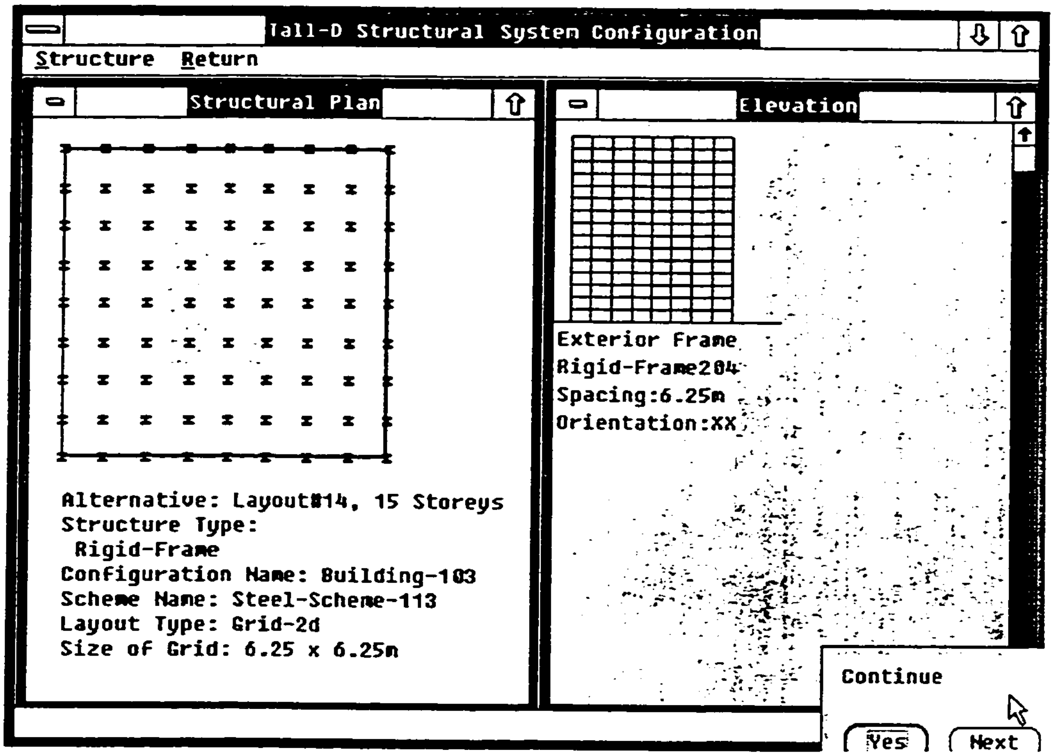


Fig. B- 7 Rigid frame strucural system alternative for Layout#14: Grid 6.25mx6.25m

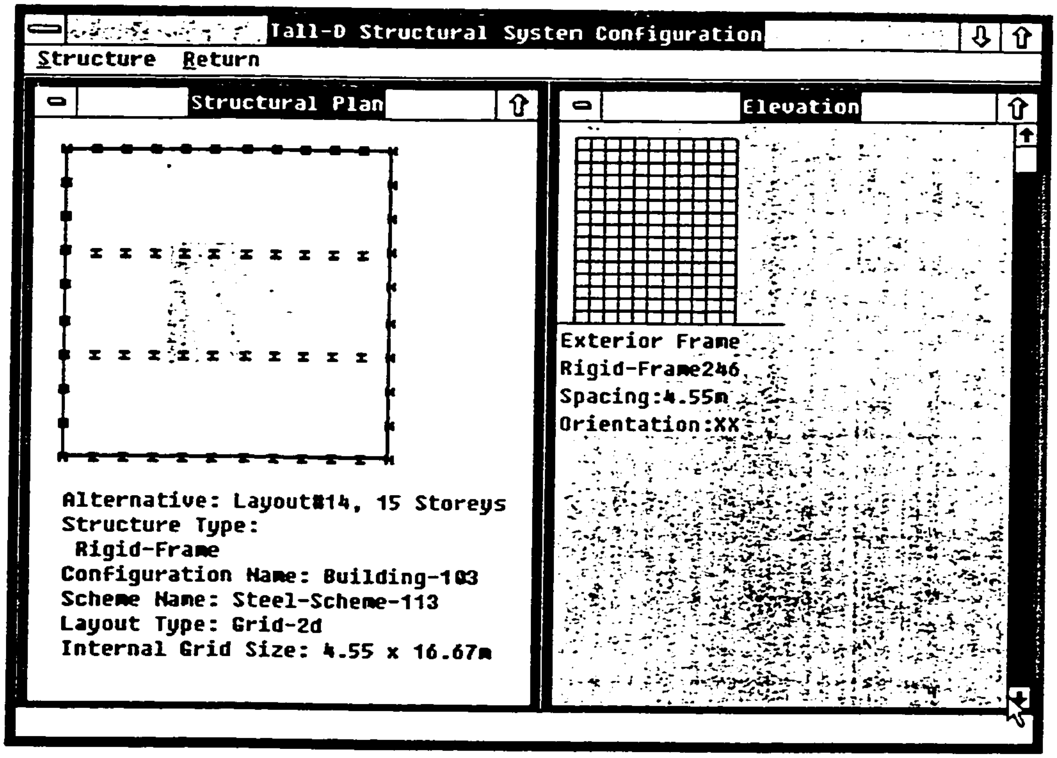


Fig. B- 8 Rigid frame strucural system alternative for Layout#14: Grid 4.55mx16.67m

Sample Images of Alternative for Layout#1

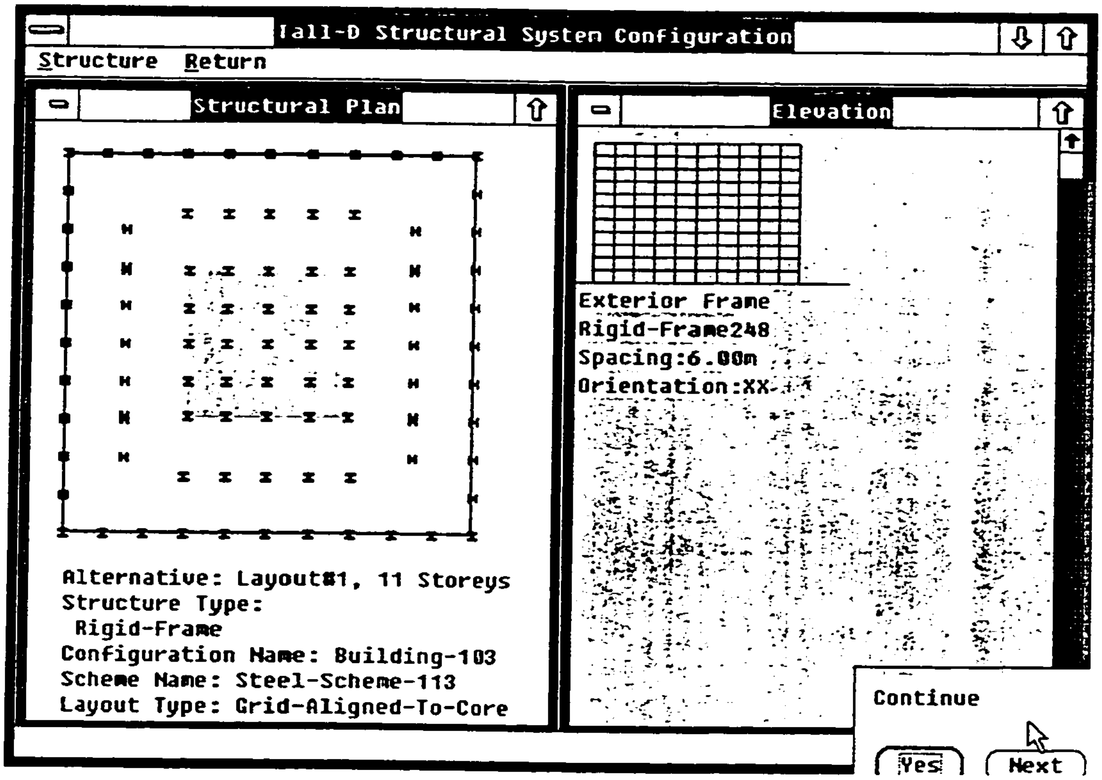


Fig. B- 9 Rigid frame strucural system alternative for Layout#1: Perimeter column spacing 6.0m

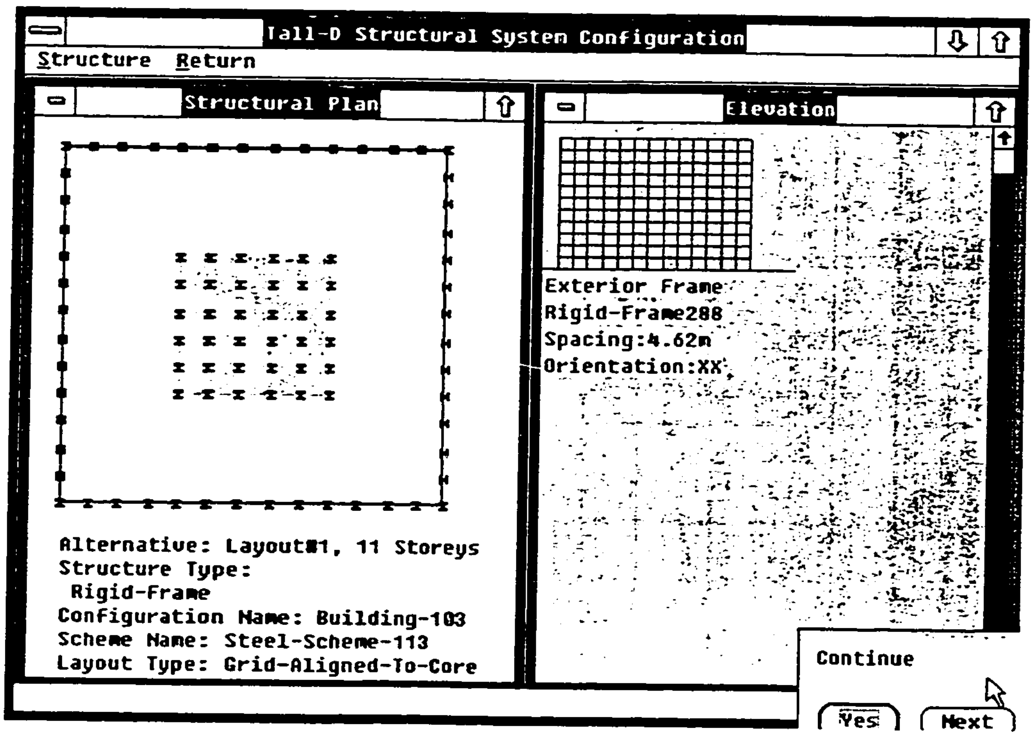


Fig. B- 10 Rigid frame strucural system alternative for Layout#1: Perimeter column spacing 4.62m

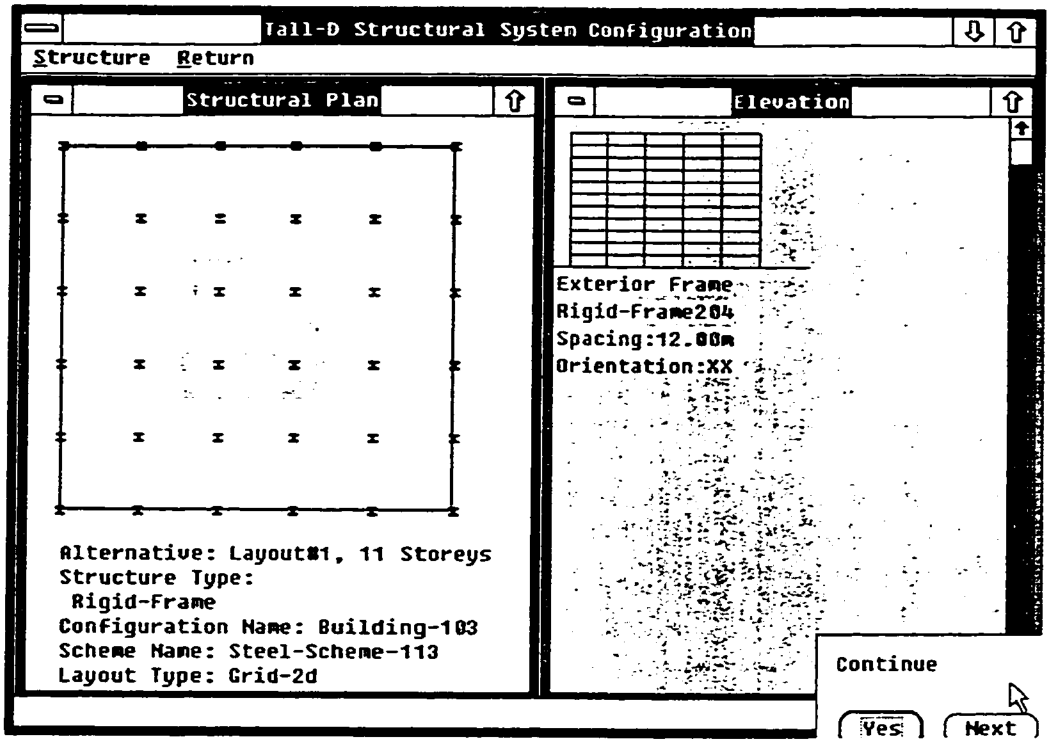


Fig. B- 11 Rigid frame strucural system alternative for Layout#1: Grid 12.0mx12.0m.

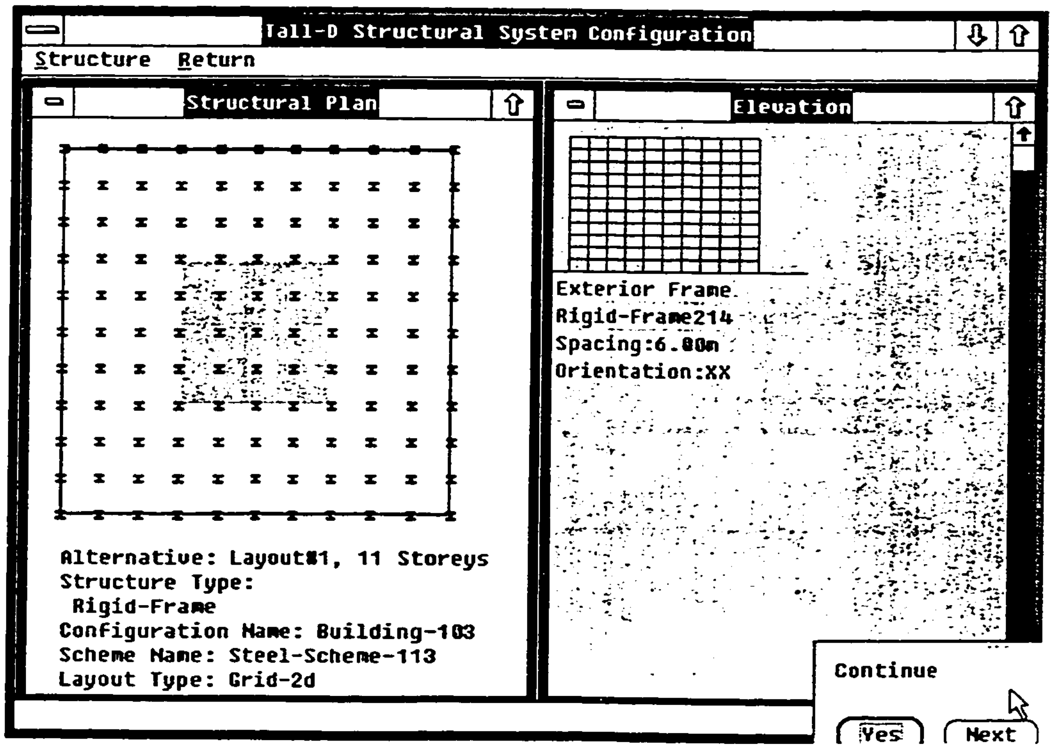


Fig. B- 12 Rigid frame strucural system alternative for Layout#1: Grid 6.0mx6.0m.

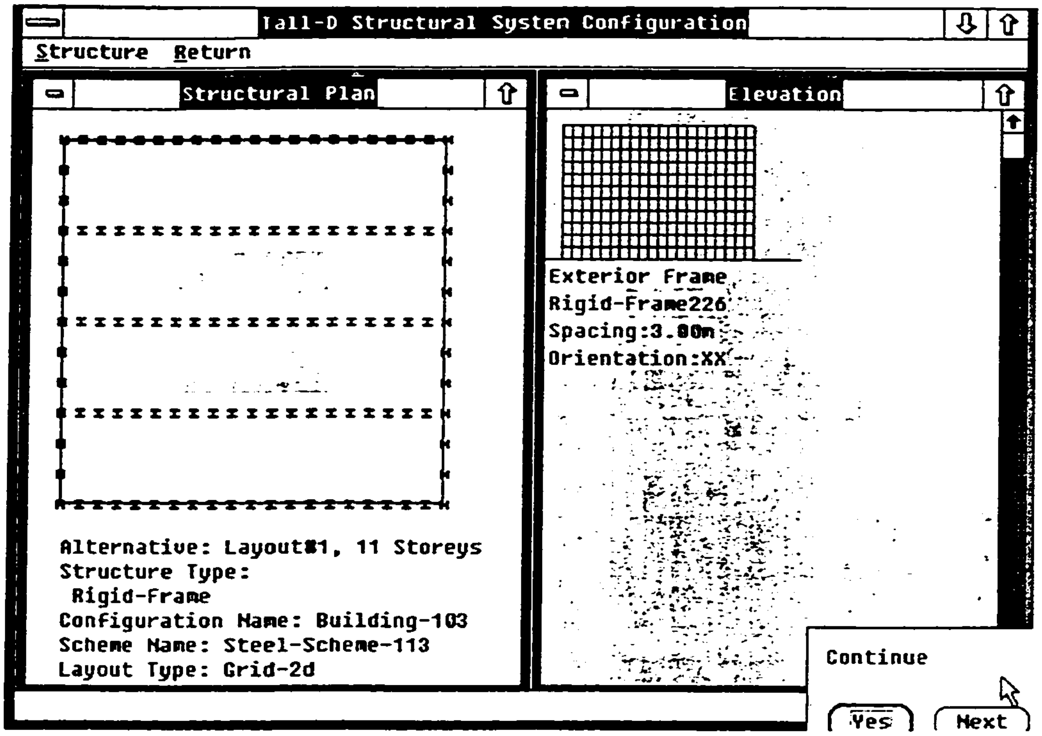


Fig. B- 13 Rigid frame strucural system alternative for Layout#1: Grid 3.00mx15.00m

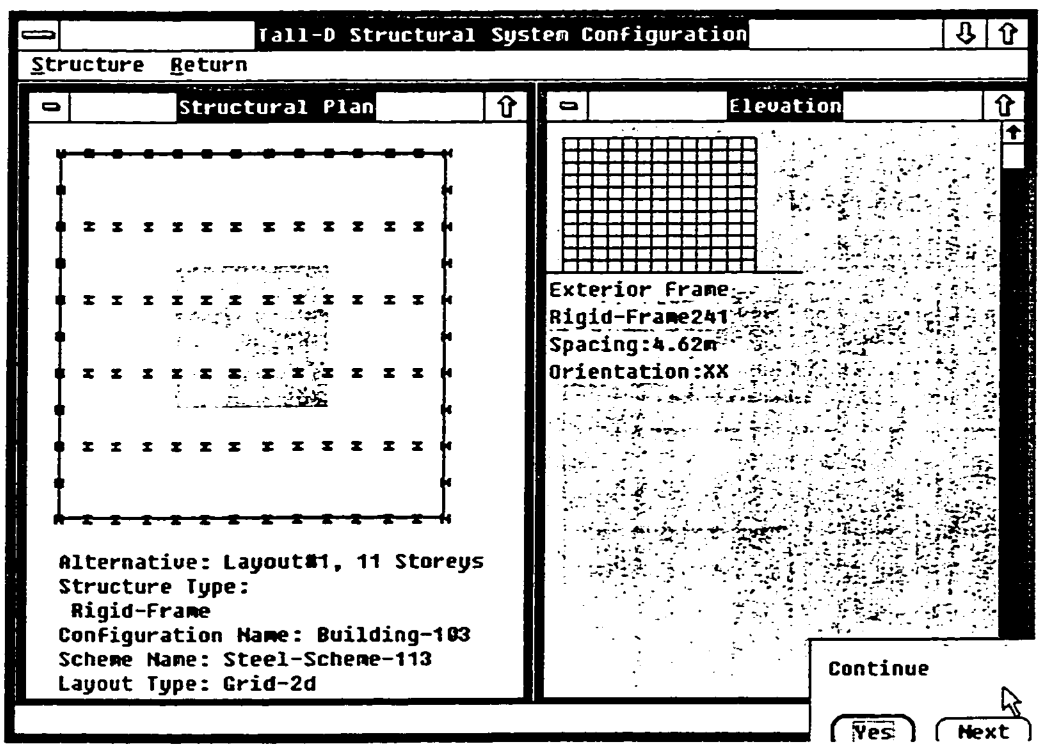


Fig. B- 14 Rigid frame strucural system alternative for Layout#1: Grid 4.62mx 12.00m

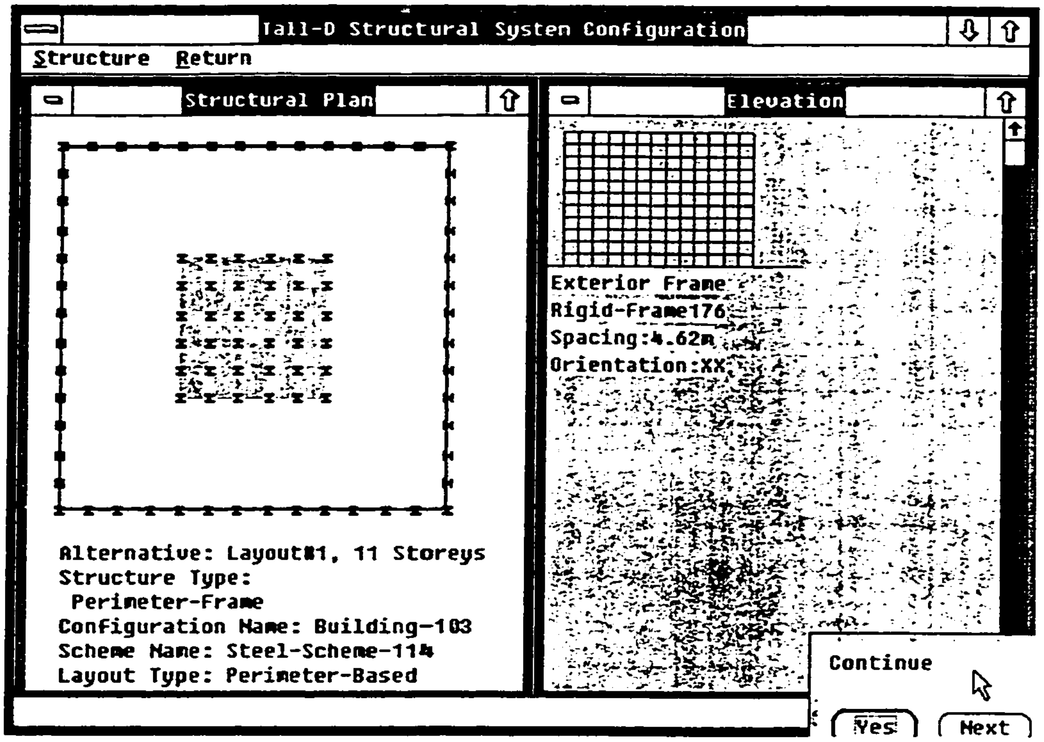


Fig. B- 15 Perimeter-based rigid frame strucural system alternative for Layout#1: Perimeter column spacing 4.62m and 4.62m

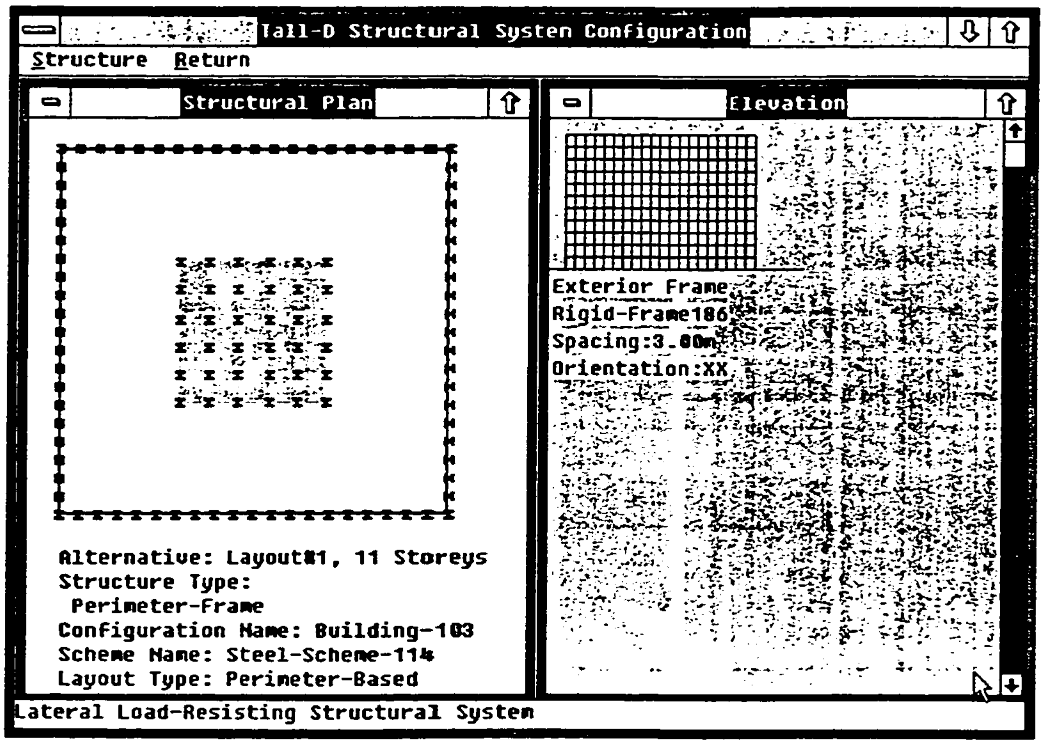


Fig. B- 16 Perimeter-based rigid frame strucural system alternative for Layout#1: Perimeter column spacing 3.0m and 3.0m

Appendix C: Contents of Knowledge-Base

Section **Page**

Appendix C-1 List of Rules in Tall-D

The following is a list of names of the approximately 415 rules in Tall-D knowledge-base The rules are grouped under different rule-sets. The list of rule-sets is presented in next section (C-2) in this Appendix. The file names shown at the top of each sublist contains the rules listed under in it. All rules in a given rule-set are related to a general task in the tall buildings design process. The numbers on the left are the line numbers in the file where the definition of the rule/function/frame begins. In Lisp coding all text that appears after the character ';' is a comment.

Initial Gravity System Rules

File Name: GC0-JU.16

Line Number	Rule Name	File Name: GC0-JU
49	Instance-Concrete-Gravity-Options	
69	Initial-Gravity-System-10	;Flat-Plate/Slab in Low seismic regn.
93	Initial-Gravity-System-12	;Low & Medium seismic regions
119	Initial-Gravity-System-14	;High seismic regions
143	Initial-Gravity-System-14.1	;High seismic regions &LargeSpan
180	Initial-Gravity-System-15	;Rigid Frame-behaviour
211	Initial-Gravity-System-20	;Concrete Frame-ShearWall
250	Initial-Gravity-System-30	;Concrete Frame-ShearWall-Haunch-Girder
285	Initial-Gravity-System-40	;Concrete Framed Tube Systems
325	Initial-Gravity-System-45	;Concrete Shear-Wall; No column
355	Initial-Gravity-System-45_1	;Concrete Shear-Wall; No column
400	Initial-Gravity-System-50	;Concrete ShearCore; No column
445	Initial-Gravity-System-50_1	;Concrete ShearCore; No column

492	Instance-Steel-Gravity-Options
512	Initial-Steel-Gravity-System-Options
549	Initial-Steel-Gravity-System-Retract ;Retract if seismicity high

Concrete Gravity Systems for Different Spans - Rules

File Name: GC1-JU.16

File Name: GC1-JU

Line Number	Rule Name
24	Concrete-gravity-Select-ShortSpanOptions-10
58	Concrete-gravity-Select-MediumSpanOptions-10
94	Concrete-gravity-Select-LargeSpanOptions-10
135	Concrete-gravity-Select-ShortSpanOptions-20
171	Concrete-gravity-Select-MediumSpanOptions-20
208	Concrete-gravity-Select-LargeSpanOptions-20
246	Concrete-gravity-Grid-2D-Equal-ShortSpanOptions-20
282	Concrete-gravity-Grid-2D-Equal-MediumSpanOptions-20
319	Concrete-gravity-Grid-2D-Equal-LargeSpanOptions-20
359	Concrete-gravity-Grid-2D-UnEqual-ShortSpanOptions-20
400	Concrete-gravity-Grid-2D-UnEqual-MediumSpanOptions-20
442	Concrete-gravity-Grid-2D-UnEqual-LargeSpanOptions-20

Steel Gravity Systems for Different Spans - Rules

File Name: GC2-JU.16

File Name: GC2-JU

Line Number	**Rule Name**
30	Steel-gravity-Select-ShortSpanOptions-10
64	Steel-gravity-Select-MediumSpanOptions-10
100	Steel-gravity-Select-LargeSpanOptions-10
141	Steel-gravity-Select-ShortSpanOptions-20
177	Steel-gravity-Select-MediumSpanOptions-20
214	Steel-gravity-Select-LargeSpanOptions-20
254	Steel-gravity-Grid-2D-Equal-ShortSpanOptions-20
290	Steel-gravity-Grid-2D-Equal-MediumSpanOptions-20
327	Steel-gravity-Grid-2D-Equal-LargeSpanOptions-20
367	Steel-gravity-Grid-2D-UnEqual-ShortSpanOptions-20
404	Steel-gravity-Grid-2D-UnEqual-MediumSpanOptions-20
442	Steel-gravity-Grid-2D-UnEqual-LargeSpanOptions-20

Concrete Slab Sizing - Rules

File Name: GC3-JU.16

File Name: GC3-JU

Line Number	**Rule Name**
19	Beam-Slab-OneWay-Sizing-10
52	Beam-Slab-TwoWay-Sizing-10
83	Flat-Plate-Sizing-10
109	Flat-Slab-Sizing-10

139	Haunch-Beam&Slab-Sizing-10
170	Band-Beam&Slab-Sizing-10
201	Concrete-Joist-Slab-Sizing-10
233	Waffle-slab-Sizing-10
260	Double-Tee-Sizing-10
293	Single-Tee-Sizing-10
325	Solid-Slab-Sizing-10
356	Hollow-Core-Slab-Sizing-10
382	Composite-Slab-Sizing-10
416	Beam-Slab-OneWay-Sizing-20
447	Beam-Slab-TwoWay-Sizing-20
478	Flat-Plate-Sizing-20
504	Flat-Slab-Sizing-20
534	Haunch-Beam&Slab-Sizing-20
565	Band-Beam&Slab-Sizing-20
596	Concrete-Joist-Slab-Sizing-20
628	Waffle-slab-Sizing-20
655	Double-Tee-Sizing-20
682	Single-Tee-Sizing-20
710	Solid-Slab-Sizing-20
741	Hollow-Core-Slab-Sizing-20
767	Composite-Slab-Sizing-20
805	Beam-Slab-OneWay-Sizing-30
836	Beam-Slab-TwoWay-Sizing-30
867	Flat-Plate-Sizing-30
893	Flat-Slab-Sizing-30
923	Haunch-Beam&Slab-Sizing-30
954	Band-Beam&Slab-Sizing-30
985	Concrete-Joist-Slab-Sizing-30

1017	Waffle-slab-Sizing-30
1044	Double-Tee-Sizing-30
1075	Single-Tee-Sizing-30
1106	Solid-Slab-Sizing-30
1137	Hollow-Core-Slab-Sizing-30
1163	Composite-Slab-Sizing-30
1198	Beam-Slab-OneWay-Sizing-40
1229	Beam-Slab-TwoWay-Sizing-40
1260	Flat-Plate-Sizing-40
1286	Flat-Slab-Sizing-40
1316	Haunch-Beam&Slab-Sizing-40
1347	Band-Beam&Slab-Sizing-40
1378	Concrete-Joist-Slab-Sizing-40
1410	Waffle-slab-Sizing-40
1437	Double-Tee-Sizing-40
1468	Single-Tee-Sizing-40
1500	Solid-Slab-Sizing-40
1531	Hollow-Core-Slab-Sizing-40
1557	Composite-Slab-Sizing-40

Steel Deck System Sizing Rules

File Name: GC4-JU.16

File Name: GC4-JU

Line Number	**Rule Name**
31	Composite-SteelBeam-Deck-Slab-Sizing-10
61	Composite-Stub-Girder-Slab-Sizing-10
92	Composite-Tapered-Beam-Slab-Sizing-10
123	Composite-Haunch-Beam-Slab-Sizing-10

154	Composite-Truss-DeckSlab-Sizing-10
185	Composite-Castellated-Beam-Slab-Sizing-10
216	Composite-Parallel-Beam-Slab-Sizing-10
253	Composite-SteelBeam-Deck-Slab-Sizing-20
283	Composite-Stub-Girder-Slab-Sizing-20
314	Composite-Tapered-Beam-Slab-Sizing-20
345	Composite-Haunch-Beam-Slab-Sizing-20
376	Composite-Truss-DeckSlab-Sizing-20
407	Composite-Castellated-Beam-Slab-Sizing-20
438	Composite-Parallel-Beam-Slab-Sizing-20
474	Composite-SteelBeam-Deck-Slab-Sizing-30
504	Composite-Stub-Girder-Slab-Sizing-30
535	Composite-Tapered-Beam-Slab-Sizing-30
566	Composite-Haunch-Beam-Slab-Sizing-30
597	Composite-Truss-DeckSlab-Sizing-30
628	Composite-Castellated-Beam-Slab-Sizing-30
659	Composite-Parallel-Beam-Slab-Sizing-30
693	Composite-SteelBeam-Deck-Slab-Sizing-40
723	Composite-Stub-Girder-Slab-Sizing-40
754	Composite-Tapered-Beam-Slab-Sizing-40
785	Composite-Haunch-Beam-Slab-Sizing-40
816	Composite-Truss-DeckSlab-Sizing-40
847	Composite-Castellated-Beam-Slab-Sizing-40
878	Composite-Parallel-Beam-Slab-Sizing-40

Semi-Rigid-Frame Rules

File Name: RL1-JU.16

Line Number	**Rule Name**	**File Name: RL1-JU**
21	Semi-Rigid-Frame-stability-element-10	
50	Semi-Rigid-Frame-stability-element-14	
76	Semi-Rigid-Frame-stability-element-16	
101	Semi-Rigid-Frame-column-spacing-30	
130	Semi-Rigid-Frame-column-spacing-31	
158	Semi-Rigid-Frame-column-spacing-32	
185	Semi-Rigid-Frame-column-spacing-33	
212	Semi-Rigid-Frame-column-spacing-33_1	
240	Semi-Rigid-Frame-column-spacing-34	
269	Semi-Rigid-Frame-column-spacing-35	
298	Semi-Rigid-Frame-40	
334	Semi-Rigid-Frame-42	
370	Semi-Rigid-Frame-44	
408	Semi-Rigid-Frame-1000	

Steel-Rigid-Frame Rules

File Name: RL2-JU.16

Line Number	**Rule Name**	**File Name: RL2-JU**
32	Steel-Rigid-Frame-10	;Including all large-span alternatives
53	Steel-Rigid-Frame-15	;Additional close-spaced perimeter frame
88	Steel-Rigid-Frame-15_1	;Additional close-spaced perimeter frame

118 Steel-Rigid-Frame-15_2 ;Additional close-spaced perimeter frame

151 Steel-Rigid-Frame-20

179 Steel-Rigid-Frame-30

211 Steel-Rigid-Frame-40

241 Steel-Rigid-Frame-EqualGrid-spacing-of-Range_4.5m ;Grid-2D Layout

280 Steel-Rigid-Frame-EqualGrid-spacing-of-Range_6.0m ;Grid-2D Layout

321 Steel-Rigid-Frame-EqualGrid-spacing-of-Range_8.0m ;Grid-2D Layout

395 Steel-Rigid-Frame-EqualGrid-spacing-of-Range_12.0m ; Grid-2D Layout

437 Steel-Rigid-Frame-EqualGrid-spacing-of-Range_15.0m ; Grid-2D Layout

476 Steel-Rigid-Frame-EqualGrid-spacing-of-Range_18.0m ;Grid-2D Layout

520 Steel-Rigid-Frame-UnEqualGrid-6x15m ;Grid-2D Layout

556 Steel-Rigid-Frame-UnEqualGrid-4.5x12m ;Grid-2D Layout

594 Steel-Rigid-Frame-UnEqualGrid-4.5x15m ;Grid-2D Layout

630 Steel-Rigid-Frame-UnEqualGrid-9x18m ;Grid-2D Layout

672 Steel-Rigid-Frame-Core-Al-Column-Layout-10

713 Steel-Rigid-Frame-Core-Al-Column-Layout-10a

757 Steel-Rigid-Frame-Core-Al-Column-Layout-10.1

799 Steel-Rigid-Frame-Core-Al-Column-Layout-10.1a

844 Steel-Rigid-Frame-Core-Al-Column-Layout-Check_Core

868 Steel-Rigid-Frame-Core-Al-Column-Layout-12 ;intermediate cols.

913 Steel-Rigid-Frame-Core-Al-Column-Layout-12a ;intermediate cols.

955 Steel-Rigid-Frame-Core-Al-Column-Layout-12.1 ;intermediate cols.

999 Steel-Rigid-Frame-Core-Al-Column-Layout-12.1a ;intermediate cols.

1043 Steel-Rigid-Frame-Core-Al-Column-Layout-20_End_Spans_Int_Column

1092 Steel-Rigid-Frame-Core-Al-Column-Layout-25

1133 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-10 ;XX

1172 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-10.1 ;YY

1212 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-Check_Core

1235 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-12 ;intermediate cols.

1278 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-12.1 ;intermediate cols.

1323 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-14 ;XX 3m

1354 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-14.1 ;YY 3m

1384 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-20_End_Spans_Int-_Column

1433 Steel-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-25

Concrete-Tubular-Frame Rules

File Name: RL3-JU.16

Line Number	**Rule Name**
27	Concrete-Tubular-Frame-stability-element-10
49	Tubular-Frame-Rule-20
79	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-10 ;XX
118	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-10.1 ;YY
157	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-Check_Core
180	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-12 ;intermed. cols.
222	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-12.1 ;intermed. cols.
267	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-14 ;XX 3m
295	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-14.1 ;YY 3m
324	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-20_End_Spans_Int_-Column
373	Tubular-Frame-Rule-Core-Al-Perimeter-Based-Col-Layout-25

File Name: RL3-JU

Braced-Frame Rules

File Name: RL4-JU.16

Line Number	**Rule Name**	**File Name: RL4-JU**
28	Braced-Frame-Rule-10 ;stability element	
48	Braced-Frame-Rule-15 ;stability element	
70	Braced-Frame-Rule-20	
96	Braced-Frame-Rule-30	
123	Braced-Frame-Rule-40	
153	Braced-Frame-EqualGrid-spacing-of-Range_4.5m ;Grid-2D Layout	
192	Braced-Frame-EqualGrid-spacing-of-Range_6.0m ;Grid-2D Layout	
233	Braced-Frame-EqualGrid-spacing-of-Range_8.0m ;Grid-2D Layout	
309	Braced-Frame-EqualGrid-spacing-of-Range_12.0m ; Grid-2D Layout	
351	Braced-Frame-EqualGrid-spacing-of-Range_15.0m ; Grid-2D Layout	
390	Braced-Frame-EqualGrid-spacing-of-Range_18.0m ;Grid-2D Layout	
434	Braced-Frame-UnEqualGrid-4.5x12m ;Grid-2D Layout	
472	Braced-Frame-UnEqualGrid-6x15m ;Grid-2D Layout	
516	Braced-Frame-Core-Al-Column-Layout-10	
558	Braced-Frame-Core-Al-Column-Layout-10.1	
601	Braced-Frame-Core-Al-Column-Layout-Check_Core	
625	Braced-Frame-Core-Al-Column-Layout-12 ;intermediate cols.	
668	Braced-Frame-Core-Al-Column-Layout-12a ;intermediate cols.	
710	Braced-Frame-Core-Al-Column-Layout-12.1 ;intermediate cols.	
754	Braced-Frame-Core-Al-Column-Layout-12.1a ;intermediate cols.	
798	Braced-Frame-Core-Al-Column-Layout-20_End_Spans_Int_Column	
847	Braced-Frame-Core-Al-Column-Layout-25	
888	Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-10 ;XX	
927	Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-10.1 ;YY	

967 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-Check_Core

990 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-12 ;intermediate cols.

1033 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-12.1 ;intermediate cols.

1078 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-14 ;XX 9m

1107 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-14.1 ;YY 9m

1137 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-20_End_Spans_Int_Column

1186 Braced-Frame-Core-Al-Perimeter-Based-Col-Layout-25

1222 Braced-Plane-Frames-1Way

1243 Braced-Plane-Frames-2Way

1266 Braced-plane-Frames-YY-1Way-Numbers ; Braced-Frames pll. to YY

1294 Braced-plane-Frames-YY-1Way-Numbers_Peri ;Braced-Frames pll.to YY

1331 Braced-Plane-Frames-XX-1Way-Numbers ; Braced-Frames pll. to XX

1362 Braced-Plane-Frames-2Way-Numbers ;Braced-Frames pll. to YY&XX Dir.: Two

1398 Braced-Frame-Rule-80 ; Braced-Frames pll. to YY

1416 Braced-Frame-Rule-90 ; Braced-Frames in perimeter

1434 Braced-Frame-Full-Width-Brace-Selection

Concrete-Rigid-Frame Rules

File Name: RL5-JU.16

Line Number	Rule Name	File Name: RL5-JU

19 Concrete-Rigid-Frame-10 ; Including all large-span alternatives

40 Concrete-Rigid-Frame-15 ;Additional close-spaced perimeter frame

75 Concrete-Rigid-Frame-15_1 ;Additional close-spaced perimeter frame

105 Concrete-Rigid-Frame-15_2 ;Additional close-spaced perimeter frame

138 Concrete-Rigid-Frame-20

166 Concrete-Rigid-Frame-30

198 Concrete-Rigid-Frame-40

228 Concrete-Rigid-Frame-EqualGrid-spacing-of-Range_4.5m ;Grid-2D Layout

267 Concrete-Rigid-Frame-EqualGrid-spacing-of-Range_6.0m ;Grid-2D Layout

308 Concrete-Rigid-Frame-EqualGrid-spacing-of-Range_8.0m ;Grid-2D Layout

382 Concrete-Rigid-Frame-EqualGrid-spacing-of-Range_12.0m ; Grid-2D Layout

420 Concrete-Rigid-Frame-EqualGrid-spacing-of-Range_15.0m ; Grid-2D Layout

459 Concrete-Rigid-Frame-EqualGrid-spacing-of-Range_18.0m ;Grid-2D Layout

499 Concrete-Rigid-Frame-UnEqualGrid-6x15m ;Grid-2D Layout

536 Concrete-Rigid-Frame-UnEqualGrid-4.5x12m ;Grid-2D Layout

574 Concrete-Rigid-Frame-UnEqualGrid-4.5x15m ;Grid-2D Layout

610 Concrete-Rigid-Frame-UnEqualGrid-9x18m ;Grid-2D Layout

652 Concrete-Rigid-Frame-Core-Al-Column-Layout-10

693 Concrete-Rigid-Frame-Core-Al-Column-Layout-10a

737 Concrete-Rigid-Frame-Core-Al-Column-Layout-10.1

779 Concrete-Rigid-Frame-Core-Al-Column-Layout-10.1a

824 Concrete-Rigid-Frame-Core-Al-Column-Layout-Check_Core

848 Concrete-Rigid-Frame-Core-Al-Column-Layout-12 ;intermediate cols.

892 Concrete-Rigid-Frame-Core-Al-Column-Layout-12.1 ;intermediate cols.

938 Concrete-Rigid-Frame-Core-Al-Column-Layout-20_End_Spans_Int_Column

988 Concrete-Rigid-Frame-Core-Al-Column-Layout-25

1028 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-10 ;XX

1066 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-10.1 ;YY

1105 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-Check_Core

1128 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-12 ;intermediate cols.

1170 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-12.1 ;intermediate cols.

1215 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-14 ;XX 3m

1243 Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-14.1 ;YY 3m

1272	Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-20_End_Spans_-Int_Column
1322	Concrete-Rigid-Frame-Core-Al-Perimeter-Based-Col-Layout-25

Frame-ShearWall Rules

File Name: RLA-JU.16

Line Number	Rule Name	File Name: RLA-JU
21	Frame-ShearWall-10	
44	Frame-ShearWall-15	
66	Frame-ShearWall-20	
88	Frame-ShearWall-30	
112	Frame-ShearWall-40 ; SHWalls pll. to YY	
148	Frame-ShearWall-50 ; SHWalls pll. to XX	
179	Frame-ShearWall-60 ; SHWalls pll. to YY & XX Dir.: Two	
214	Frame-ShearWall-80 ; SHWalls pll. to YY	
231	Frame-ShearWall-90 ;Composite Structure? Steel Columns?	
260	Frame-ShearWall-Core-Al-Column-Layout-10	
309	Frame-ShearWall-Core-Al-Column-Layout-10a	
361	Frame-ShearWall-Core-Al-Column-Layout-10.1	
408	Frame-ShearWall-Core-Al-Column-Layout-10.1a	
456	Frame-ShearWall-Core-Al-Column-Layout-11	
504	Frame-ShearWall-Core-Al-Column-Layout-11a	
556	Frame-ShearWall-Core-Al-Column-Layout-11.1	
602	Frame-ShearWall-Core-Al-Column-Layout-11.1a	
651	Frame-ShearWall-Core-Al-Column-Layout-12 ;intermediate cols.	
703	Frame-ShearWall-Core-Al-Column-Layout-12a ;intermediate cols.	
753	Frame-ShearWall-Core-Al-Column-Layout-12.1 ;intermediate cols.	

805	Frame-ShearWall-Core-Al-Column-Layout-12.1a ;intermediate cols.
857	Frame-ShearWall-Core-Al-Column-Layout-20_End_Spans_Int_Column
911	Frame-ShearWall-Core-Al-Column-Layout-25

Steel-Framed-Tube Rules

File Name: RLD-JU.16

Line Number	Rule Name		File Name: RLD-JU
20	Steel-Framed-Tube-10		
41	Steel-Framed-Tube-15		
63	Steel-Framed-Tube-20		
85	Steel-Framed-Tube-30		
109	Steel-Framed-Tube-40	; SHWalls pll. to YY	
145	Steel-Framed-Tube-50	; SHWalls pll. to XX	
176	Steel-Framed-Tube-60	; SHWalls pll. to YY & XX Dir.: Two	
211	Steel-Framed-Tube-80	; SHWalls pll. to YY	
228	Steel-Framed-Tube-90	;Composite Structure? Steel Columns?	
257	Steel-Framed-Tube-Core-Al-Column-Layout-10		
306	Steel-Framed-Tube-Core-Al-Column-Layout-10a		
358	Steel-Framed-Tube-Core-Al-Column-Layout-10.1		
405	Steel-Framed-Tube-Core-Al-Column-Layout-10.1a		
454	Steel-Framed-Tube-Core-Al-Column-Layout-11		
505	Steel-Framed-Tube-Core-Al-Column-Layout-11a		
557	Steel-Framed-Tube-Core-Al-Column-Layout-11.1		
602	Steel-Framed-Tube-Core-Al-Column-Layout-11.1a		
650	Steel-Framed-Tube-Core-Al-Column-Layout-12 ;intermediate cols.		
703	Steel-Framed-Tube-Core-Al-Column-Layout-12a ;intermediate cols.		
754	Steel-Framed-Tube-Core-Al-Column-Layout-12.1 ;intermediate cols.		

806	Steel-Framed-Tube-Core-Al-Column-Layout-12.1a ;intermediate cols.
856	Steel-Framed-Tube-Core-Al-Column-Layout-13 ; 3m perim.col.spc.
903	Steel-Framed-Tube-Core-Al-Column-Layout-13a ; 3m perim.col.spc.
954	Steel-Framed-Tube-Core-Al-Column-Layout-13.1 ; 3m perim.col.spc.
999	Steel-Framed-Tube-Core-Al-Column-Layout-13.1a ; 3m perim.col.spc.
1047	Steel-Framed-Tube-Core-Al-Column-Layout-20_End_Spans_Int_Column
1101	Steel-Framed-Tube-Core-Al-Column-Layout-25

Overall Building Configuration Rules

File Name: RUL-JU.16

File Name: RUL-JU

Line Number	**Rule Name**
17	Actv-Generate-Floor-Rules
36	Actv-Evalaute-Floor-Rules
54	Actv-Approx-Bldg-Costs
78	Display-Intro
94	Ask-Owner-Reqr
117	Display-Status
141	Deactv-user-interface-rules
169	Max-floor-area
194	GEN-FP1
223	GEN-FP2
251	Write-Out-1
269	Write-out-2
287	Aspect-Ratio-Bldg
314	Slenderness-Characteristic
336	Put-in-Cores
358	Check-no-of-Floors

379	Geometry-of-core
410	Deact-Gen-Flrs
433	Eval-Floors-AsPer-Priority
470	Assign-Ranks-to-Flrs
488	Write-floor-priority
523	Deactv-Evaluate-Floor-rules
545	Find-Approximate-cost
563	Approx-Struct-Material
584	Find-Relative-Structural-cost
603	Deactv-Approx-Cost

Construction Material Rules

624	Construction-Material-10
639	Construction-Material-15
657	Construction-Material-15.1
674	Construction-Material-15.2
693	Construction-Material-15.3
710	Construction-Material-15.4
728	Construction-Material-15.5
745	Construction-Material-20
761	Construction-Material-22
785	Construction-Material-25
803	Construction-Material-27
820	Construction-Material-30
837	Construction-Material-40
855	Construction-Material-50
874	Construction-Material-60

Structure Type and Window-line Rules

899	Structure-Type-Concrete-10	
916	Structure-Type-Concrete-15	; For Detached Cores
935	Structure-Type-Concrete-15_1	; For Detached Cores
957	Structure-Type-Concrete-17	; For Edge Cores
975	Structure-Type-Concrete-20	
996	Structure-Type-Concrete-25	
1013	Structure-Type-Concrete-30	
1033	Structure-Type-Concrete-32	
1053	Structure-Type-Concrete-33	
1069	Structure-Type-Concrete-35	
1089	Structure-Type-Concrete-40	
1108	Structure-Type-Concrete-50	
1131	Structure-Type-Steel-10	
1151	Structure-Type-Steel-15	
1168	Structure-Type-Steel-20	
1190	Structure-Type-Steel-30	
1209	Structure-Type-Steel-40	
1226	Structure-Type-Steel-45	
1244	Structure-Type-Steel-50	
1265	Structure-Type-Dueto-Slenderness-10	
1288	Structure-Type-Dueto-Slenderness-20	
1308	Structure-Type-Dueto-Slenderness-30	
1332	Structure-Type-Dueto-Slenderness-40	
1362	Core-to-WindowLine-Central-core	
1394	Core-to-WindowLine-Edge-core	
1425	Core-to-WindowLine-Detached-core	
1454	Core-to-WindowLine-End-cores	

1471 Max-Pos-Span-End-cores

1497 Core-to-WindowLine-Two-core

1529 Core-to-WindowLine-Corner-type

Rule-Sets Activation Rules

1577 Actv-Semi-Rigid-Frame-Config

1595 Actv-Staggered-Truss-Config

1612 Actv-Steel-Rigid-Frame-Config

1631 Actv-Braced-Frame-Config

1648 Actv-Belt-Truss-with-Braced-frame-Config

1664 Actv-Framed-End-Channel-Config

1681 Actv-Framed-Tube-Steel-Config

1700 Actv-Concrete-Rigid-Frame-Config

1717 Actv-ShearCore-Config

1734 Actv-Shear-Wall-Config

1750 Actv-Frame-ShearWall-Config

1767 Actv-Frame-ShearWall-Haunch-Girder-Config

1783 Actv-Framed-Tube-Concrete-Config

Miscellaneous (not fully devoloped modules) Rules

File Name: RL6-JU.16

20 Belt-Truss-with-Braced-frame-10

File Name: RL7-JU.16

20 Framed-End-Channel-10

File Name: RL8-JU.16

25 Shear-Wall-10

47 Shear-Wall-20 ;SWall-Direction

69 Shear-Wall-30 ;SWall-Direction

File Name: RL9-JU.16

20 Staggered-Truss-10

File Name: RLC-JU.16

25 Frame-ShearWall-Haunch-Girder-10

Appendix C-2 List of Rule-Sets in Tall-D

The following is a name list of 25 rule-sets that are used to group the rules in Tall-D system. Each set can be activated and deactivated from within the application so as to restrict the inference engine to the relevant rules at any time. It also helps to repeatedly use any given set of rules if need be. On a practical side the rule-sets helped speed up the system and helped avoid an insufficient memory situtation.

Line Number	Rule Set Name	File Name: RUL-JU
1854	User-Interface-Rules	
1866	Generate-floor-Rules	
1885	Evaluate-Floor-Rules	
1897	Approximate-Cost-Rules	
1908	Construction-Material-Rules	
1932	Structure-Type-Selection-Rules	
1964	Geometric-Information-Rules	
1982	Activate-Structure-SubModule-Rules	
2007	Semi-Rigid-Frame-Configuration-Rules ()	
2029	Steel-Rigid-Frame-Configuration-Rules()	
2074	Staggered-Truss-Configuration-Rules ()	
2082	Braced-Frame-Configuration-Rules ()	
2131	Belt-Truss-with-Braced-frame-Configuration-Rules ()	
2140	Framed-End-Channel-Configuration-Rules ()	
2149	Concrete-Rigid-Frame-Configuration-Rules ()	
2198	Shear-Wall-Configuration-Rules ()	
2207	ShearCore-Configuration-Rules ()	
2215	Frame-ShearWall-Configuration-Rules ()	
2245	Frame-ShearWall-Haunch-Girder-Configuration-Rules ()	
2256	Framed-Tube-Concrete-Configuration-Rules ()	
2276	Framed-Tube-Steel-Configuration-Rules ()	

2311	Generate-Gravity-System-Options-Rules
2339	Concrete-Gravity-System-Rules
2362	Concrete-Gravity-Scaling-Rules
2424	Steel-Gravity-System-Rules
2445	Steel-Gravity-Scaling-Rules

List of Sponsors in Tall-D

Sponsors in the GoldWorks KBS tool are used to allocate inference engine resources. They can be assigned rule-sets as well as individual rules. They are *enabled* and *disabled* to include and exclude respectively the rules they are assigned. There are some similarities and differences between sponsors and rule-sets. Where sponsors are a higher level control of the system resources, rule-sets enable control of smaller group of rules, especially when that group of rules needs to be forced through the inference engine due to the design context. Sponsors on the other hand fire a group of rules only once, and to do otherwise one has to go through elaborate reallocation of resources.

File name: SPO-JU.16

File Name: SPO-JU.16

Line Number	**Frame/Object Name**
17	FLOOR-CONFIGURATION
24	USER-INTERFACE
30	Generate-Floor
37	Cost-Approximation
44	Evaluate-Floor
50	Structure-Configuration
57	Gen-Lateral-System
63	Construction-Material
69	Gen-Gravity-System
77	Envelope-Configuration

Appendix C-3 List of Frames in Tall-D

The following 110 frames (or objects) are used in hybrid knowledge-representation in Tall-D.

File name: FRM-JU.16

Line Number	**Frame/Object Name**	**File Name: FRM-JU**
17	BUILDING-OBJECT	
40	Building	
93	Building-Configuration	
120	Generic-System	
129	Super-Structure	
168	Floor-Outline	
199	Constraints	
205	Code-constr	
219	Owner-Constr	
243	Floor-System-Options	
255	Steel-Floor-System-Options	
297	Concrete-Floor-System-Options	
339	Building-Loads	
372	Core-Config	
396	Design-Status	
400	FL-GENERATION-STATUS	
411	Structure-Design-Status	
437	General-constraints	
467	Eval-Criteria	
510	Rigid-Frame-Scheme	
534	Concrete-Rigid-Frame-Scheme	
543	Steel-Rigid-Frame-Scheme	
553	Braced-Frame-Scheme	

Line Number	**Frame/Object Name**
575	Shear-Wall-Scheme
596	ShearCore-Scheme
606	Frame-ShearWall-Scheme
642	Frame-ShearWall-Haunch-Girder-Scheme
652	Belt-Truss-with-Braced-Frame-Scheme
661	Staggered-Truss-Scheme
671	Framed-Tube-Scheme
709	Semi-Rigid-Frame-Scheme
717	Semi-rigid-scheme-perimeter-frame
735	Semi-rigid-scheme-Int-Braced-frame
754	Semi-rigid-scheme-Ext-Braced-frame
775	Framed-End-Channel-Scheme
782	Perimeter-Braced-frame-scheme
804	Lateral-Structural-System
843	Gravity-Structural-System
857	Exterior-Closure
863	Sub-Structure
875	Generic-Element
884	Structural-Element
898	Architectural-Element
905	Beam-Element
912	Wall-Element
919	Column-Element
937	Slab-Element
946	Structural-Image
956	Frame-Moments
965	BEAM-IMAGE
974	Column-IMAGE

Line Number	**Frame/Object Name**
982	Wall-IMAGE
990	Slab-IMAGE
997	Concrete-Beam
1012	Concrete-Haunch-Beam
1023	Steel-Beam
1034	Composite-Beam ; Steel rolled section encased in concrete
1041	Stub-Girder
1049	Tapered-Beam
1058	Steel-Haunch-Beam
1066	Castellated-Beam
1075	Parallel-Beam
1085	Concrete-Column
1103	Steel-Column
1119	Composite-Column
1127	ShearWall-Element
1151	Concrete-Slab
1158	Steel-Slab
1165	Composite-Deck-Slab
1176	Lateral-System-Image
1185	Gravity-System-Image
1198	Framed-Tube
1209	Framed-End-Channel
1221	ShearCore
1235	Frame-ShearWall
1253	Frame-ShearWall-Haunch-Girder
1267	Rigid-Frame
1281	Braced-Frame
1304	Semi-Rigid-Frame

Line Number	**Frame/Object Name**
1315	Staggered-Truss
1325	Belt-Truss-with-Braced-Frame
1339	Column-Layout
1434	Beam-Slab-OneWay
1449	Beam-Slab-TwoWay
1464	Haunch-Beam&Slab ;Concrete
1478	Flat-Slab
1491	Flat-Plate
1502	Band-Beam&Slab
1515	Concrete-Joist-Slab
1530	Waffle-slab
1543	Double-Tee ;may be with solid or Hollow core & concrete topping
1556	Single-Tee ;may be with solid or Hollow core & concrete topping
1570	Solid-Slab
1582	Hollow-Core-Slab
1594	Composite-Slab ; CONCRETE == Precast+In-Situ combination
1608	Composite-SteelBeam-Deck-Slab
1623	Composite-Stub-Girder-Slab
1638	Composite-Tapered-Beam-Slab
1654	Composite-Haunch-Beam-Slab
1670	Composite-Truss-DeckSlab
1686	Composite-Castellated-Beam-Slab
1701	Composite-Parallel-Beam-Slab
1724	Last-Outline

Frames Used to Setup CISC Steel Tables

The following eight frames were used to setup the CISC steel tables as a set of instances in Tall-D. Due to the limited system resources as well as the limited use of these tables in the preliminary stage of design they were later disabled in the system.

File name: PARSE.LSP

Line Number	Frame/Object Name
153	Steel-Structural-Section
189	W-Section
195	S-Section
201	M-Section
207	HP-Section
213	WWF-Section
219	WRF-Section
227	C-Section

File Name: PARSE.LSP

List of Predefined Instances in Tall-D

The following 30 predefined instances are used in Tall-D for such functions as popping a question or displaying graphics. However most of the instances in Tall-D are generated during execution of the system.

File name: INS-JU.16

Line Number	Instance Name
23	Owner-Constraints
31	Building-Loads-Code
39	Code-Constraints

File Name: INS-JU.16

Appendix C-4 List of Structural Design Lisp Routines

The following 78 Lisp routines perform some of the structural computations discussed in the text of the thesis in sections 5.1 through 5.3. Extracted from source code file STR-JU, there are two types of functions. Those in the list below that begin with a parenthesis, are handler functions attached to objects, also briefly described below. The others are functions that are not specific to objects but are task oriented. The same format is followed in Appendix sections C-4 through C-7.

Handler functions are methods attached to frames/objects. They are also known as member functions of the object. These functions have access to the dynamically instanced object data members or slot-values. The first set of parentheses declares the name of the object and handler function. The second set of parenthesis encloses optional parameters. These attached functions are activated by sending a message to the owner object, which could be done from the rule-base or any other lisp code in the system.

File Name: STR-JU

Line Number	**Function Name**
26	(Building-Configuration Generate-Loads-on-the-Building) ()
53	(Building-Configuration Generate-Lateral-Loads) ()
103	(Building-Configuration Generate-Gravity-Loads)()
221	Generate-Column-Tributary-Areas (Layout)
489	Put-tributaryArea-in-Rigid-FrameXXrYY-if-Grid2D
534	Call-Approx-Analysis-Function()
561	Get-Schemes(Bldg All-Schemes)
577	Send-message-to-Schemes-for-Analysis()
622	Display-Choice-Submenu-of-Str-to-Analyse (All-Schemes)
676	(ShearWall-Element Find-Wall-Thickness&Properties) ()
865	(Column-Layout Size-Columns-for-Gravity-for-all-Alternative-ColSpacings) ()
974	DelCol(Inst)
1008	(Concrete-Column Size-Section) (Type InitRFinal)

1071	Get-Initial-Col-Size (Tributary-area No-of-Storeys Type Strength)
1133	Get-Variable-Size-of-ColumnStack-Along-BldHgt
1202	Minimun_300mm(Concrete-Col-Side)
1211	Con-Col-Rationalise (Size)
1230	(Steel-Column Size-Section) (Type InitRFinal)
1290	Get-Initial-Steel-Col-Size (Tributary-area No-of-Storeys Type Strength)
1342	Get-Variable-Size-of-Steel-Column-Along-BldHgt
1391	Tabulate-Column-Sections-for-All-Alternatives(Stream)
2166	Weight (Designation)
2191	Cal-Shear (Shear-Incr N-storeys)
2206	(Rigid-Frame Rigid-Frm-Moment) ()
2258	T-Mmt ()
2265	Del-Mmt ()
2272	(Rigid-Frame Draw-Moments) (Window)
2330	Draw-Moment (Window Sxi Szi Bay-Size M Sc Ht)
2350	FindCon-BEAM-ELEMENT-DEPTH
2428	BeamSize-Beam-Slab-OneWay
2436	BeamSize-Beam-Slab-TwoWay
2444	BeamSize-Concrete-Joist-Band ;Site-cast
2451	BeamSize-Band-Beam&Slab
2461	BeamSize-Haunch-Beam&Slab
2470	BeamSize-Single-Tee
2479	BeamSize-Double-Tee
2501	Con-Beam-Rationalise (Size)
2516	Con-BandBeam-Rationalise (Size)
2532	Con-Slab-Rationalise (Size)
2551	Find-SLAB-ELEMENT-DEPTH
2624	SlabSize-Beam-Slab-OneWay
2706	One-Way-Slab-thick (Span)

2713	SlabSize-Beam-Slab-TwoWay
2797	Two-Way-Slab-thick (Span)
2805	SlabSize-Flat-Plate
2893	Flat-Plate-Slab-thick (Span)
2901	SlabSize-Flat-Slab
2989	Flat-Slab-thick (Span)
2995	Find-Drop-thickness&Width
3008	SlabSize-Concrete-Joist-Slab
3089	Joist-Slab-Rib-Depth (Span)
3098	SlabSize-Waffle-Slab
3183	Waffle-Slab-thick (Span)
3197	SlabSize-Band-Beam&Slab
3291	SlabSize-Haunch-Beam&Slab
3379	SlabSize-Hollow-Core-Slab
3440	Hollow-Core-Slab-thick (Span)
3451	SlabSize-Single-Tee
3521	SlabSize-Double-Tee
3592	FindSteel-Beam-Element-Depth
3643	Steel-Beam-Rationalise (Size)
3658	BeamDWComposite-SteelBeam-Deck-Slab
3682	BeamDWComposite-Stub-Girder-Slab
3703	BeamDWComposite-Tapered-Beam-Slab
3725	BeamDWComposite-Haunch-Beam-Slab
3748	BeamDWComposite-Truss-DeckSlab
3766	BeamDWComposite-Castellated-Beam-Slab
3788	BeamDWComposite-Parallel-Beam-Slab
3814	Find-DeckSlab-Element-Depth
3826	Print-Driver-for-Gravity-System ()
3841	Print-Driver-for-Gravity-System-Subroutine(Scheme)

3937 Print-Gravity-system-Instance-Details(Inst &OPTIONAL Stream)

3989 Get-Relevant-Grav-Instances(Scheme All-Grvs)

4007 Belongs-to-This(Scheme Grav-Inst)

4023 Order-Gravity-Instances-As-Per-Expected-Relative-Cost (Scheme)

4054 Order-Gravity-Alts-Based-On-Cost-Using-Span(Control-Span ConcrRSteel)

Appendix C-5 List of DXF and CISC Interface Lisp Routines

The following 8 lisp routines in Tall-D implement a DXF interface to AutoCad.

File name: ACD-JU.16

Line Number	Function Name
29	Tall-D-to-AcadDXF (Building Scheme Col-Layout)
162	Put-FramesXXYY-in-3D ()
269	ACADSTART()
278	ACAD-STOP()
292	MOVE-CURRENT (x y z)
300	SET-CURRENT (x y z)
309	LINE3DFRMTO (x1 y1 z1 x2 y2 z2)
321	LINE3DTO (x2 y2 z2)

List of CISC Interface Routines

File name: PARSE.LSP

Line Number	Function Name
47	make-parse-descriptor ( )
96	Indentify-properties-Rec1 (LS) ;LS List of all records from the file

Appendix C-6 List of Building Configuration Lisp Routines

The following 113 functions perform different general purpose tasks in Tall-D as the descriptive names of the functions suggest in most instances. These lines are extracted from file FUN-JU.16.

File Name: FUN-JU

Line Number	**Function Name**
120	(floor-outline Pass-values&Draw) (len wid name)
132	(Last-Outline Draw) (Window)
159	draw-the-outline (x y)
182	Alternate-draw-the-outline (x y)
216	rec-repaint (window)
258	Display-View-FP (window item)
271	confirm-info (template)
286	set-max-bldg-len (instance slot old-value new-value)
297	Cal-Area-per-floor-with-detached-core&Revise-no-of-floors
335	set-max-bldg-wid (instance slot old-value new-value)
347	Do-initial-check (instance slot old-value new-value)
366	Set-*PRCT*-Variable (instance slot old-value new-value)
380	(Owner-Constr Check-Finance-Limit)
411	generate-floor-dim-ranges (Reqr-Fl-Ar)
443	set-all-slot-value (ins slot lst)
451	(floor-outline set-floor-area-ratio) (XL YL PLOT-X Plot-Y)
462	set-value-for-FAR (instance value)
472	eliminate-infeasible-FAR (fo PL-X PL-Y) ; PL: Plot dimensions
505	calculate-no-of-floors (instance)
531	eliminate-too-tall (inst slot old-val new-val)
561	comp-flrs-ifn() ;compute no. of floors if necessary
576	Put-To-File-1 ()

Line Number	Function Name
605	put-to-file-2 ()
622	Filter-BSR-&to-file-3 ()
643	put-to-file-4 ()
667	put-to-file-4.1 ()
692	Print-Core-Config (Floor-ALt stream)
712	Print-Designer-Priority-Flrs (
740	Print-Evaluation-Info()
793	cal-bldg-asp-ratio (Ins No-of-fl fl-length fl-width fl-height Stat)
815	set-bldg-slenderness (Ins Bar N)
832	pick-compatible-cores (Floor-Alt FL FW Class)
876	set-core-dimensions (floor-alt Fl-length fl-width core-type
1005	draw-with-core (x y cl1 cw1 cx1 cy1 type FA &OPTIONAL First-Layout)
1089	Repaint-Fl+Cores (window x1 y1 wid higt)
1112	Set-Storey-Global-Variable (inst slot old-vals new-vals)
1116	Find-Building-Height (inst slot old-vals new-vals)
1187	(Column-Layout Check-Columns-in-Core) (StrSch PerimColSpX PerimColSpY)
1239	(Braced-Frame-Scheme Do-Braces-Location-TypeSelection) (NBraces SpcX Location Dir)
1469	(Frame-ShearWall-Scheme Create-SHWall-Instances) (NWalls-YY SpcX Location Direction)
1573	Get-the-Right-Layout (All-Layts)
1592	Set-Flange-Length (inst slot old-values new-values)
1609	(Framed-Tube-Scheme Create-SHWall-Instances) (NWalls-YY SpcX Location Direction)
1712	(Super-Structure Initiate-Create-Plane-Frame-Instances)()
1760	(Column-Layout Provide-Geometry-for-Lateral-System) (SteelRConcrete)
2105	(Super-Structure Generate-One-Plane-Frame) (Plane-Frame-Type XXrYY

Line Number	Function Name
	Material X Y SpcCol NCol Displ FlPlan ColLayt Special_Cond)
2197	Find-Type&Size-of-Bracing (Instance Slot Old-Values New-Values)
2280	Start-User-Interface (window item)
2298	Reset-All&Start (window item)
2310	Generate-floor-plans (window item)
2367	Find-if-Any-Layouts-Survived() ;;And display-appropriate message
2403	Evaluate-Floor-Plans(window item)
2417	Initiate-Approximate-cost (window item)
2436	Activate-Structure-Type-Selection (window item)
2651	delete-Redundant-Floor-Alts()
2658	Close-Str-Scrn-Layout&OpenMain (Window Item)
2668	Change-Displayed(Inst)
2677	Activate-Gravity-Module (Window Item)
2730	Display-Ask-for-Gravity-Generation-Schemes ()
2783	Display-Submenu-of-Structure-Choices-for-Gravity ()
2804	Initiate-Assemble-Building()
2827	(Structure-Design-Status Activate-Each-SubModule) ()
2845	Display-Prompt (instance slot Old-value New-value)
2864	Sys-Get-Floors()
2882	Get-Num-of-Floors (instance slot Old-value New-value)
2901	Display-Floor-Choice()
2932	Building-Combos-Available (Lst)
2968	(General-Constraints Find-Floor&Core-Instances) ()
3005	Floor-Eval-with-Priority
3041	Vect-Mult (A B Result)
3050	Assign-Ranks-to-Floors()
3081	Get-rank-range (TN Ordered-Evs Rank-range)
3094	Set-Initial-Ranks (Cr-List)

Line Number	Function Name
3102	Display-Finish-Eval ()
3163	Get-Live-Load-factors (inst slot old-vals new-vals)
3177	Display-Evaluation-Info (window item)
3216	Display-Structure-Loading-Options (window item)
3249	Make-Strings(Lst)
3266	Generate-Lateral-System-Instances(Load-R-Disp)
3326	Pair-XXRigid&YYBraced-Frames()
3395	Pair-XX-with-YY-Frames()
3449	VerifyBraceBays-NonCentral-Core-Grid2D()
3513	Evaluate-Str-Alts(window item)
3522	Display-Structure-Evaluation-Criteria ()
3534	Do-Approximate-StrAnalysis (window item)
3565	Structural-alts-Generated-p()
3576	Loads-Generated-p ()
3625	Display-Message ()
3635	Do-Approximate-Member-Sizing (window item)
3648	Show-Current-Alternatives (window item)
3669	Make-List-of-Structures(Lst)
3689	Ask-Save-File&Save-StructureDesign (window item)
3700	Ask-for-Floor-Modifications (window item)
3711	Floor-Modify-NePas()
3731	Ask-for-Floor-Selection (window item)
3747	Display-View-Str(window item)
3800	Display-Submenu-of-Structure-Choices ()
3819	List-of-LateralSys()
3867	ConvertTo-Strings(Lst)
3883	Display-View-Str-Elevations (window item)
3945	(Structure-Design-Status Send-Message-to-Str-Schemes) (Schemes)

Line Number	Function Name
3952	(Structure-Design-Status Send-Message-to-Str-Schemes-Elevation) (Schemes)
3970	(Floor-outline Find-Approximate-Cost) ()
3979	Total-Cost (Instance)
4022	Cost-Index (City)
4036	City-Correction(City Natnl-Avg)
4049	(Core-Config Effect-of-Storeys-on-Struct-Material) ()
4065	Relative-Structural-Approx-Cost ()
4100	Display-Submenu-of-Schemes-4-Relative-Cost()
4117	Display-the-cost-info (window item)
4178	Report-Str-Alts(&OPTIONAL File-Stream)
4222	Report-Scheme (Str-Sch)
4440	msg()

Function that initiates the Tall-D system:

File name: LOD-JU.16

63 run-tall-builder ()

The following 6 functions were mostly used during initial development of a skeletal system by trapping the menu item selections in the Tall-D user interface and displaying a message to the effect that.

File Name: TRK-JU.16

41 Temp-Str-GenLoads ()

68 Temp-Eval-StrAlter ()

90 Temp-Approx-StrAnalysis ()

112 Temp-Approx-StrSizing ()

139 Temp-Save-StrDesign ()

167 Temp-Floor-Select ()

Appendix C-7 List of Graphical Lisp Routines

The following 27 functions perform mostly Windows graphical displays by drawing the structural elevations and plan views with values accessed from the instances representing the design alternatives. Graphics is initiated mainly by the structural design function and rules.

File Name: WIN-JU.16

Line Number	Function Name	File Name: WIN-JU
53	start-Tall-D (&optional First-P)	
69	START-BUILDER (&optional (restart nil))	
77	Initialise-Main-Windows ()	
332	(Rigid-frame draw) (Window)	
391	Rel-Move-X (Xi Zi X-Inc)	
397	Rel-Move-Z (Xi Zi Z-Inc)	
407	(Frame-ShearWall draw) (Window)	
491	(Braced-Frame draw) (Window)	
588	Draw-Brace (Window Type x z s B2)	
605	One-decimal (Lst)	
610	trim (No)	
618	Place-Column-Y (Sxi Syi X Y) ; Sxi: x along c/l; syi: y at left of c/l	
646	Place-Column-X (Sxi Syi X Y) ;(X,Y) x,y displacements	
678	Place-ConCol-X (Sxi Syi X Y)	
693	Place-ConCol-Y (Sxi Syi X Y)	
714	(Super-Structure Display-Structure-Elevation) ()	
785	(Super-Structure Display-Structure) ()	

Line Number	Function Name
850	(Super-Structure Draw-Outline) (Window Col-Layout-Type Stability-Element X Y)
919	Put-One-Elevation-on-Canvas (All-FrmsX-List Spcx Layout)
1006	(Column-Layout Draw-Columns-in-Plan&Elevation-for-ALL-One-Variation-at-a-Time) (Window SteelRConcrete)
1327	Draw-Cols-Along-XX (CTyp Sxi Syi Spacing Ncol-X Y)
1346	Draw-Cols-Along-YY (CTyp Sxi Syi Spacing Ncol-Y X)
1374	Draw-All-Cols-in-Grid(CTyp Xt Yt Xg Yg SpcX SpcY NBays NAisls Window)
1397	(Column-Layout Draw) (Window)
1428	(Frame-Shearwall-Scheme Add-SHWall-Images-to-Plan) (Window X Y)
1467	(ShearWall-Element Draw) (Window)
1555	(Framed-Tube-Scheme Add-SHWall-Images-to-Plan) (Window X Y)

Appendix C-8 Lisp Routines for Initial Sizing of Columns

Part-1: Concrete Columns

```
;;C O N C R E T E:  Columns
;;
;; This handler does initial sizing by using tributary area for
;; short columns and height of column for slender columns.
;; Since in the beginning no column section size is available.
;; Hence the higher of the two sizes used.
;; Side effect: Grouping of storeys for same column outer dimensions
;; slender or short
;; slender: slenderness of column has to be considered; complicated one:
;;          usually encountered in abnormally large floor to floor height
;;           & sometimes where the load is light and need to make the
;;           col. slender for economy; but economy not guaranteed
;;           due to more stringent requirements for slender columns
;;           However in high-rise blds., columns are loaded heavily
;;           which ensures that the size arrived at by tributary area
;;          is more that that arrived at by using only height of column.
;; short:   (kl/r) of column is low enough to neglrct slenderness effects
;;           design is much simpler for this category. Initial size is
;;          by chart from Allen & Iano for col. size with tributary area.
;;
(define-handler (Concrete-Column Size-Section) (Type InitRFinal)
     ;Type:slender/short InitRFinal:Initial/Final
  (Let* (
    (Frm (slot-value Self 'Part-of))
    (Layt (slot-value Frm 'Part-of))
    (Scheme (slot-value Layt 'Related-Structural-Scheme))
    (Bld (slot-value Scheme 'Part-of))
    (StoreyHt (slot-value Bld 'Storey-Height))
    (Tributary-area (slot-value Self 'Tributary-area-per-floor))
    (No-of-Storeys (slot-value Bld 'Number-of-Floors))
    (Concrete (slot-value Self 'Compressive-strength)) ; in MPa
    (No-storeys-in-Each-column-group
      (Cond
   ((<= No-of-Storeys 10) 2)
   ((<= No-of-Storeys 20) 3)
   ((<= No-of-Storeys 30) 3)
   ((<= No-of-Storeys 40) 4)
   ((> No-of-Storeys 40) 5)
   (t 5)
   )
      )
    (Number-of-Groups
      (If (NOT (= (Mod No-of-Storeys No-storeys-in-Each-column-group) 0))
     (floor (+ 1 (/ No-of-Storeys No-storeys-in-Each-column-group)))
   (floor (/ No-of-Storeys No-storeys-in-Each-column-group))))
    (Last-group-Nos (mod No-of-Storeys No-storeys-in-Each-column-group))
    (Col-Siz 0) ;  temp. variable for column size
    )
    (Case InitRFinal
      ('Initial
   (setf Col-Siz
        (Get-Initial-Col-Size Tributary-area No-of-Storeys
```

```
              Type Concrete))
    )
       ;; Final Get-Final-Col-Size function to use values from
       ;; lateral load anal. plus gravity load/critical combination
       ('Final
    (setf Col-Siz (Get-Final-Col-Size Tributary-area No-of-Storeys
               Type Concrete))
    )
       )
     ;; Increase by 50% to account for lateral load
     (setf Col-Siz (* (sqrt 1.5) Col-Siz))
     ;; Rationalise size to nearest 50mm
     (setf Col-Siz (Con-Col-Rationalise Col-Siz))
     (format t "~&COl-Siz #stry in Group #groups last.grp.nos ~D ~D ~D ~D"
        Col-Siz   No-storeys-in-Each-column-group
        Number-of-Groups Last-group-Nos)
     (set-slot-value Self 'Number-of-Storeys-in-Typical-Group
          No-storeys-in-Each-column-group)
     (set-slot-value Self 'Number-of-Groups Number-of-Groups)
     (set-slot-value Self 'Number-of-Storeys-in-Last-Group Last-group-Nos)
     (set-slot-value Self 'Side-of-Square-Column-at-Base Col-Siz)
     )
   )



; function to set initial column size based on tributary area
; live load reduction factor applied here!
; concrete strength effect in reducing or increasing is applied
;          (Based on Allen & Iano table)
; called by above function
(defun Get-Initial-Col-Size (Tributary-area No-of-Storeys Type Strength)
  (Let* (
    (Concrete-Strength Strength)
    (Gross-TA (* Tributary-area No-of-Storeys))
    (Redu-Factor (+ 0.3 (sqrt (/ 9.8 Tributary-area))))
    (Redu-TA (* Redu-Factor Gross-TA)) ; reduced tributary area
    (Col-Size-for35MPa
      (If (not-equal Type 'Slender)
     ;; The following is the side of square section of concrete column
     ;; Ref. Allen & Iano
     ;; Since these sizes are to be increased (approx 50%) for
     ;; lateral load smallest section begin with 200 mm sqr.
     ;; to finally endup with 300mm columns
     (Cond ((<= Redu-TA 200) 200)  ;mm side of sqaure section
      ((AND (> Redu-TA 200) (<= Redu-TA 350))  250)
      ((AND (> Redu-TA 350) (<= Redu-TA 450))  350)
      ((AND (> Redu-TA 450) (<= Redu-TA 600))  450)
      ((AND (> Redu-TA 600) (<= Redu-TA 700))  550)
      ((AND (> Redu-TA 700) (<= Redu-TA 750))  600)
      ((AND (> Redu-TA 750) (<= Redu-TA 900))  650)
      ((AND (> Redu-TA 900) (<= Redu-TA 1100)) 700)
      ((AND (> Redu-TA 1100) (<= Redu-TA 1250))  750)
      ((AND (> Redu-TA 1250) (<= Redu-TA 1400))  800)
      ((> Redu-TA 1400)  850)
      )
```

```
300)) ; Initialize slender columns to 300mm
  ;; correction for higher or lower strength than 35MPa concrete
  ;;    i.e. chart used is for 35MPa concrete
  ;; (based on Allen & Iano)
  (Modification
    (Cond
((= Concrete-Strength 35) 1.0)
((<= Concrete-Strength 20) 1.25)
((<= Concrete-Strength 25) 1.15)
((<= Concrete-Strength 30) 1.05)
((<= Concrete-Strength 35) 1.)
((<= Concrete-Strength 40) .97)
((<= Concrete-Strength 45) .96)
((<= Concrete-Strength 50) .95)
((<= Concrete-Strength 55) .87)
((<= Concrete-Strength 60) .82)
((<= Concrete-Strength 65) .78)
((<= Concrete-Strength 70) .76)
((<= Concrete-Strength 75) .75)
((> Concrete-Strength 75) .75)
)
   )
  )
  (float (/ (floor (* Col-Size-for35MPa Modification)) 1000)) ; mm->m
  ;;;(format t "~& Need routine for slender column initial sizing")
  )
 )



;;;When-modified function for
;; Parent frame: Concrete-Column
;;          slot: Side-of-Square-Column-at-Base
;;
;; Final Output: Section at different heights of the building
;; Each section:(Initial area derived by gravity load) * 1.5 to account
for
;;               lateral loads as first approximation
(Defun Get-Variable-Size-of-ColumnStack-Along-BldHgt
       (Inst Slot old-values new-values)
  (format t "~&Yes I'm Working ~A ~A " Old-values New-values)
  (Let* (
    (Concrete (slot-value Inst 'Compressive-strength))
    (Gross-Trib-Area-per-floor (slot-value Inst
                               'Tributary-area-per-floor))
    (No-of-Groups (slot-value Inst 'Number-of-Groups))
    (Group-Increment (slot-value Inst
                         'Number-of-Storeys-in-Typical-Group))
    (Column-Stack-Size-List '())
    (Redu-Fac 0)
    (Flr-No-from-top (slot-value Inst 'Number-of-Storeys-in-Last-Group))
    (Flr-No-from-top (If (= Flr-No-from-top 0)
         Group-Increment Flr-No-from-top))
    (Type 'Initial)
    (Current-Section 0)
    )
```

```
    (Dotimes (I (- No-of-Groups 1))
      (setf Current-Section
    (Get-Initial-Col-Size Gross-Trib-Area-per-floor Flr-No-from-top
                Type Concrete)
    )
      (format t "~& Current-Section ~A" Current-Section)
      (setf Column-Stack-Size-List
       (cons Current-Section Column-Stack-Size-List))
      (setf Flr-No-from-top (+ Flr-No-from-top Group-Increment))
      )
    ; Add column at the base already obtained
    (setf Column-Stack-Size-List
     (cons (car new-values) Column-Stack-Size-List))
    ;;
    ;;*** Increase column sizes obtained by 50% at bottom 1/3rd
    ;;    and 50% (same increase - could be less at the base)
    ;;    for rest of columns to account for lateral load ***
    ;;    Smith & Coull method
    (let* (
      (third (- (floor (/ No-of-groups 3.3)) 1))
      (rest (- No-of-groups third))
      (lst1 '(1.0))
      (lst2 '())
      (Incr-Lst '())
      )
      (Dotimes (I third)
  (setf lst1 (cons (sqrt 1.5) lst1))
  )
      (setf lst1 (reverse lst1))
      (Dotimes (I rest)
  (setf lst2 (cons (sqrt 1.5) lst2))
  )
      (setf Incr-Lst (merge 'List lst1 lst2 'NULL))
      (format t "~& Increase List: ~A" Incr-Lst)
      (format t "~& Column-Stack-Size-List ~A: " Column-Stack-Size-List)
      (setf Column-Stack-Size-List
       (mapcar #'* Incr-Lst Column-Stack-Size-List))
      ;; Check if any of the columns are less than 300mm in size
      (setf Column-Stack-Size-List
       (mapcar #'Minimun_300mm Column-Stack-Size-List))
      (format t "~& Column-Stack-Size-List ~A: " Column-Stack-Size-List)
      ;; make sizes in 50mm inrements
      (setf Column-Stack-Size-List
       (mapcar #'Con-Col-Rationalise Column-Stack-Size-List))
      )
   (format t "~& Column-Stack-Size-List ~A: " Column-Stack-Size-List)
   (set-slot-value
     Inst 'Column-Sections-Along-Height Column-Stack-Size-List)
   )
 )


;; fn. used by above fn.
(defun Minimun_300mm(Concrete-Col-Side)
  (If (< Concrete-Col-Side 0.300)
      0.300
```

```
    Concrete-Col-Side)
  )

;; fn used by Get-Variable-Size-of-ColumnStack-Along-BldHgt
;; Rounds off column to nearest 50mm
;; returns the value in m
(Defun Con-Col-Rationalise (Size)
  (Let* (
    (mmSize (* 1000 Size))
    (Rational-mmSize
      (If (> (Mod mmSize 50) 25)
     (* 50 (+ 1 (floor (/ mmSize 50))))
   (* 50 (floor (/ mmSize 50)))
   ))
    )
    (float (/ Rational-mmSize 1000)) ; mm -> m
    )
  )
```

Part-2: Steel Columns

```
;; STEEL COLUMN INITIAL SIZING
;; Similar to the corresponding Steel handler
;; to size steel columns using initially, tributary area,
;; and subsequently to checked with lateral load member forces
(define-handler (Steel-Column Size-Section) (Type InitRFinal)
     ;Type:slender/short InitRFinal:Initial/Final
  (Let* (
    (Frm (slot-value Self 'Part-of))
    (Layt (slot-value Frm 'Part-of))
    (Scheme (slot-value Layt 'Related-Structural-Scheme))
    (Bld (slot-value Scheme 'Part-of))
    (StoreyHt (slot-value Bld 'Storey-Height))
    (Tributary-area (slot-value Self 'Tributary-area-per-floor))
    (No-of-Storeys (slot-value Bld 'Number-of-Floors))
    (Steel-Str *Steel-Compressive-Strength*) ; in MPa
    (No-storeys-in-Each-column-group
      (Cond
   ((<= No-of-Storeys 10) 2)
   ((<= No-of-Storeys 20) 3)
   ((<= No-of-Storeys 30) 3)
   ((<= No-of-Storeys 40) 4)
   ((> No-of-Storeys 40) 5)
   )
      )
    (Number-of-Groups
      (If (NOT (= (Mod No-of-Storeys No-storeys-in-Each-column-group) 0))
     (floor (+ 1 (/ No-of-Storeys No-storeys-in-Each-column-group)))
   (floor (/ No-of-Storeys No-storeys-in-Each-column-group))))
    (Last-group-Nos (mod No-of-Storeys No-storeys-in-Each-column-group))
    (Col-Siz '()) ;  temp. list for column nom. size & flange width
    )
    (Case InitRFinal
      ('Initial
```

```
   (setf Col-Siz
         (Get-Initial-Steel-Col-Size Tributary-area No-of-Storeys
               Type Steel-Str))
   )
      ;; Final Get-Final-Steel-Col-Size function to use values from
      ;; lateral load anal. plus gravity load/critical combination
      ('Final
   (setf Col-Siz (Get-Final-Steel-Col-Size Tributary-area No-of-Storeys
                Type Steel-Str))
   )
      )
    ;; Increase by 50% to account for lateral load
   ;(setf Col-Siz (Get-50%-Bigger-Steel-Col Col-Siz)) ;Increases by 50%
for LL
    (format t "~&Steel:COl-Siz #stry in Group #groups~
       last.grp.nos ~A  ~D  ~D  ~D"
       Col-Siz   No-storeys-in-Each-column-group
       Number-of-Groups Last-group-Nos)
    (set-slot-value Self 'Number-of-Storeys-in-Typical-Group
         No-storeys-in-Each-column-group)
    (set-slot-value Self 'Number-of-Groups Number-of-Groups)
    (set-slot-value Self 'Number-of-Storeys-in-Last-Group Last-group-Nos)
    (setf (slot-all-values Self 'Nominal-Size-of-Column-at-Base) Col-Siz)
    )
  )

; function to set initial column size based on tributary area
; live load reduction factor applied here!
; called by above function
(defun Get-Initial-Steel-Col-Size (Tributary-area No-of-Storeys Type
Strength)
  (Let*(
    (Steel-Strength Strength)
    (Gross-TA (* Tributary-area No-of-Storeys))
    (Redu-Factor (+ 0.3 (sqrt (/ 9.8 Tributary-area))))
    (Redu-TA (* Redu-Factor Gross-TA)) ; reduced tributary area
    ;; Size for gravity only
    (Nominal-Col-Size
      (If (not-equal Type 'Slender)
     ;; Nominal size & flange of section of Steel column
     ;; Using the tributary area and chart from
     ;; Ref. Allen & Iano, (and extending it)
     ;; These sizes are to be increased (approx 50%) for
     ;; moment commected frames contributing to lateral load
     ;; resistence. This portion is for short columns.
     ;; All columns assumed short to begin with.
     (Cond ((<= Redu-TA 80) '(W150 150))
      ((AND (> Redu-TA 80) (<= Redu-TA 230))  '(W200 200))
      ((AND (> Redu-TA 230) (<= Redu-TA 300)) '(W250 250))
      ((AND (> Redu-TA 300) (<= Redu-TA 830)) '(W310 310))
      ((AND (> Redu-TA 830) (<= Redu-TA 1200))  '(W360 360))
      ((AND (> Redu-TA 1200) (<= Redu-TA 1600)) '(W460 280))
      ((AND (> Redu-TA 1600) (<= Redu-TA 2300)) '(W610 325))
      ((AND (> Redu-TA 2300) (<= Redu-TA 3900)) '(WWF400 400))
      ((AND (> Redu-TA 3900) (<= Redu-TA 4600)) '(WWF550 550))
                            ((> Redu-TA 4600)  '(WWF800 400))
```

```
      )
   '(W250 250))) ; Initialize slender columns to W250
     ;;
     ;; correction for higher  strength steel (350MPa) can be applied
     ;; if the column capacity is not suff. but don't want increased
     ;; column size; Not used at present
     (Modification
       (If (> Steel-Strength 340)
      (format t "~&Can reduce memb. size or can take incr. load")))
     )
    Nominal-Col-Size
     )
  )



;; To use the varying tributary area along the column height
;; to find an initial size for the column
;;*When-modified function* for
;; Parent frame: Steel-Column
;;         slot: Nominal-Size-of-Column-at-Base
;;
;; Final Output: Section at different heights of the building
;; Each section:(Tributary-area) * 1.5 to account for
;;               lateral loads as first approximation
;;               in Rigid-frames and 1.25 (ie. 25%) in Braced-Frames
(Defun Get-Variable-Size-of-Steel-Column-Along-BldHgt
       (Inst Slot old-values new-values)
  (format t "~&Steel Column: Old ~A New ~A " Old-values New-values)
  (Let* (
    (Steel-Str (slot-value Inst 'Compressive-strength))
    (Partof (slot-value Inst 'Part-of))
    (FramTyp
      (If (eql (instance-Parent Inst) (gw-object 'Braced-Frame))
        'Braced-Frame
     'Rigid-Frame))
    (Gross-Trib-Area-per-floor (slot-value Inst
        'Tributary-area-per-floor))
    (Augmented-Trib-Area (* Gross-Trib-Area-per-floor
           (If (eql FramTyp 'Braced-Frame)
         1.25 1.5)))
    (No-of-Groups (slot-value Inst 'Number-of-Groups))
    (Group-Increment (slot-value Inst
             'Number-of-Storeys-in-Typical-Group))
    (Column-Stack-Size-List '())
    (Redu-Fac 0)
    (Flr-No-from-top (slot-value Inst 'Number-of-Storeys-in-Last-Group))
    (Flr-No-from-top (If (= Flr-No-from-top 0)
         Group-Increment Flr-No-from-top))
    (Type 'Initial)
    (Current-Section 0)
    )
    (Dotimes (I (- No-of-Groups 1))
      (setf Current-Section
       ;; Use augmented tributary area to get initial section
   (Get-Initial-Steel-Col-Size Augmented-Trib-Area Flr-No-from-top
```

```
            Type Steel-Str)
  )
    (format t "~& Current-Section ~A" Current-Section)
    (setf Column-Stack-Size-List
     (cons Current-Section Column-Stack-Size-List))
    (setf Flr-No-from-top (+ Flr-No-from-top Group-Increment))
    )
  ; Add column at the base already obtained
  (setf Column-Stack-Size-List
   (cons new-values Column-Stack-Size-List))
  (format t "~& Column-Stack-Size-List ~A: " Column-Stack-Size-List)
  (set-slot-value
    Inst 'Column-Sections-Along-Height Column-Stack-Size-List)
  )
)
```

Appendix C-9 Lisp Routines for Sizing of One-Way Solid Slab

The first of the following three functions scans the column layout of the current alternative to determine the overall dimensions of the bay, then based on the proportions of the bay, divides it further to make it a two-way spanning slab. Since one-way slab is always used within a slab span of 5m to 6m, it is always possible to make a grid with girders and beams to make a one way slab system. When the span is greater than 6m, this function splits the aisle span into two by an intermediate beam, which results in a one-way slab spanning perpendicular to the core to window-line span.

The second of the functions in this section returns the depth of the one-way slab, given the span. It is the straight line form representation of the chart used for this sizing procedure based on span (Allen and Iano 1989).

The third and final function presented, rounds off the value given to the nearest 10mm and is called by the previous sizing function.

```
;; One way Slab thickness
(Defun SlabSize-Beam-Slab-OneWay
       (Grav-Sys-Inst MaxSpan Part-of-List Slab-Type Slot-Name)
  ;Part-of-List = (Schm Layt Span1 Span2)
  ;Eg. Slab-Type = Beam-Slab-OneWay
  ;Eg. Slot-Name = ClearSpan-Gravity-System
                  ;(slot in Column-Layout Inst.) OR
                ;  = TwoSpans-Gravity-System
                ;  = UnEqualSpan-Gravity-Systems
                ;  = EqualSpan-Gravity-Systems
  (Let*(
    (Inst Grav-Sys-Inst)
    (Layt (Second Part-of-List))
    (Layt-Type (slot-value Layt 'Type))
    (PeriSpcYY  (slot-all-values Layt 'Perimeter-Col-Spacings-Y))
    (PeriSpcXX  (slot-all-values Layt 'Perimeter-Col-Spacings-X))
    )
    ;; If Layt is fr.shwal or perimeter-based or tubular based
    ;; or grid aligned to core then use perimeter column spacing
    ;; for finding the short span (span of slab)
    ;; For Grid-2D use equal span, unequal span grid values
    (Cond
      ((OR (Equal Layt-Type 'Perimeter-Based)
      (Equal Layt-Type 'Tubular-Based)
      (Equal Layt-Type 'Core-Centered)
      (Equal Layt-Type 'As-per-ShearWalls)
      (Equal Layt-Type 'Grid-Aligned-to-Core))
       ;; Use the grater of (XX YY) perimeter column spacing
       ;;
       ;; Alternative perim. column spacing; diff. slab thickness
```

```
        ;; BUT ALL stored in one instance as a single value slot
        ;; as ((SpanPeri1 thick1 No.ofIntermediteBeams)
        ;;      (SpanPeri2 thick2 No.ofIntermediteBeams)...)
        (Let*(
    (Alternative-Spc&thikns '()) ;Initialize
    (Count 0)
     )
    (Dolist (EachSpcX PeriSpcXX)
      (Setf Spc (If (< EachSpcX (Car (Nthcdr Count PeriSpcYY)))
           (Car (Nthcdr Count PeriSpcYY))
               EachSpcX))
      (setf count (+ 1 Count))
      (setf Split (If (> Spc 6) 2 1))
      (setf Thick (One-Way-Slab-thick (/ Spc Split)))
      (setf Alternative-Spc&thikns
       (cons (list Spc Thick Split)
             Alternative-Spc&thikns))
      )
    (format t "~&One Way Slab:~&  ~A" Alternative-Spc&thikns)
    Alternative-Spc&thikns )        )

      ;; Layout is Grid-2D
      (t
       (Let*(
    ; Find where in the list of grav sys in this slot value
    (Whole-List (slot-all-values Col-Layt Slot-Name))
    (Numbr (Position Inst Whole-List))
    (Grid-Size
      (If (Equal Slot-Name 'UnEqualSpan-Gravity-Systems)
           (car (Nthcdr Numbr 'Unequal-Grid-Spacings))
         ;;Otherwise EqualSpan layout
          (List (car (Nthcdr Numbr 'Bay-Spacings))
           (car (Nthcdr Numbr 'Aisle-Spacings)))
          ))
    (SlabSpan (car Grid-Size))
    (NSplits (If (> SlabSpan 6)
            (Floor (/ SlabSpan 3)) ;No. of int.beams 3m one way slab
         1))  ; else no intermediate beams
    (ActualSpan (/ SlabSpan NSplits))
    (Slab-Thk (One-Way-Slab-thick ActualSpan)) ;Call function
    (Output-List (List ActualSpan Slab-Thk NSplits))
    )
    (format t "~&OneWay Slab: ~A" Output-List)
    Output-List
    )
        )
      )
    )
  )

;; Used by above function to find oneway slab thickness
(Defun One-Way-Slab-thick (Span)
  ;;Allen & Iano p.113
```

```
  (Con-Slab-Rationalise (/ (- (* 49.03 Span) 15.67) 1000))
  )


;; Round off to 10mm
;; receive value in mm & return in m
(Defun Con-Slab-Rationalise (Size)
  (Let* (
    (mmSize (* 1000 Size))
    (Rational-mmSize
      (If (> (Mod mmSize 10) 5)
     (* 10 (+ 1 (floor (/ mmSize 10))))
   (* 10 (floor (/ mmSize 10)))
   ))
    )
    (float (/ Rational-mmSize 1000)) ; mm -> m
    )
  )
```

Appendix C-10 Lisp Routines for Lateral Loads and Portal Method of Analysis

```
;; This function generates for the building
;; lateral loads according to NBC of Canada 1990
;; Approximations/simplifications are noted where applicable
;; External pressure/suction P = qCeCqCp
;; q =  [=0.37 kPa for Dorval 1/30 prob]
;; Ce exposure factor, value from table 4.1.8.A (0.9 to 2.0) based on H
;; Cg gust factor
;; Cp external pressure coeff. averaged over the? area
;;
(define-handler (Building-Configuration Generate-Lateral-Loads)
     ()
  (Let*(
     (Height (slot-value Self 'Building-Height))
     (q (slot-value 'Building-Loads-Code 'Hourly-wind-pressure))
     (H23 (/ Height 1.5)) ; 2 thirds height of building
     (Ce (expt (floor (/ H23 10.0)) 0.2)) ;Ce at top third
                                ;height of building
       ;actually varies along the height of building.
      ;To get continuous var
       ;use the above expression,
      ;substitute H23 by the corresp. elevation(m)
     (Ce (Cond
      ((< Ce 0.9) 0.9)
      (t Ce))
    )
     (Cg 2.0) ;for buildings as a whole;
           ;2.5 for elements eg. cladding etc.
     (CpWin 0.8) ; Cp on wind-ward side
     (CpLee 0.5) ; Cp on lee-ward side
     ;;(P_Win_H1 (* q Ce Cg CpWin)) Not used
     (P_Win_Abv_H1 (* q Ce Cg CpWin)) ; Currently: Uniform distribution
                    ;top-third value assume uniform due to approx. the
                    ; same effect caused at the base.
                    ;
                    ; Use Ce from table 4.1.8A as per Z - elevation
                    ; of point on the building, if
               ;continuous var. required.
                    ;
     (P_Lee (* q Ce Cg CpLee))
     )
    (Print (list "P_Win_Abv_H1 P_Lee (in kPa):"  P_Win_Abv_H1 P_Lee))
    (set-slot-value Self
           'Windward-Pressure-at-Top3dHeight P_Win_Abv_H1) ;kPa
    (set-slot-value Self 'Leeward-Pressure-at-Top3dHeight P_Lee)  ;kPa
    )
  )


;; Routine Used in portal method
(defun Cal-Shear (Shear-Incr N-storeys)
  (progn
    (setq *STORY-SHR-LIST* (list (* 0.5 shear-incr)))
    (setq half-shear (* 0.5 Shear-incr))
```

```
    (Dotimes (N N-storeys)
      (setq *story-shr-list* (Append *STORY-SHR-LIST*
         (list (+ half-shear (* Shear-Incr (+ N 1))))))
      (print *story-shr-list*)
      )))



;;
;; Portal method of rigid frame analysis.
;  Valid for buildings about 25 storeys
;  This fn generates a list of form
;   ( (...)(...) (...) ) for colmn moments
(Define-handler (Rigid-Frame Rigid-Frm-Moment) ()
       (Let* (
    (N-Bays (- (Slot-value self 'number-of-Columns) 1))
                          ;; No. of Columns = N-bays + 1
    (Bay-dimensions (slot-value Self 'Column-Spacings))
    (N-Storeys (Slot-value (Slot-value Self 'Floor-Outline-Instance)
          'no-of-floors))
    (storey-shear-list *story-shr-list*)
    (Storey-Ht 4)
    (Lat-Load-per-stry 36.8) ; in kN
    )
    (Setq Total-bay (mapcar #'+ Bay-dimensions))
    (setq 1st-Col-pro
        (/ (car bay-dimensions) 2.0 Total-bay))
    (setq proptns-lst (list 1st-Col-pro))
              ;shr 1st column; modf 6.5
    (Dotimes (Bay-i (- N-Bays 1))
         (Setq Span1 (car (nthcdr Bay-i Bay-Dimensions)))
         (Setq Span2 (car (nthcdr (+ 1 Bay-i) Bay-Dimensions)))
         (setq proportion (/ (+ span1 span2) 2.0 Total-bay))
         (setq proptns-lst (append proptns-lst (list Proportion))))
       (setq last-col-pro
        (/ (car (last bay-dimensions)) 2 Total-bay))
       (setq proptns-lst (append proptns-lst (list last-col-Pro)))
       ;(print Proptns-lst)
       (setq list-of-lists-Col-Moments '())
       ;(print list-of-lists-Col-Moments)
       (setq Haf-storey-Ht (* 0.5 Storey-Ht))
       (Dotimes (Bay-i (+ 1 N-bays))
         (Setq Ratio (car (Nthcdr Bay-i Proptns-lst)))
         (setq lateral-col-moments '())
         (Dotimes (storey-i N-storeys)
      (setq storey-shear(car
             (nthcdr storey-i *story-shr-list*)))
      (setq lateral-col-shear (* ratio storey-shear))
      (Setq lateral-col-moments;top storey value first in list
            (append lateral-col-moments
          (list (* Haf-storey-Ht lateral-col-shear)))))
        ;(Print lateral-col-moments)
        (setq list-of-lists-Col-Moments ;leftmost col. first
```

```
        (append list-of-lists-Col-Moments
           (list lateral-col-moments)))
          ;(Print list-of-lists-Col-Moments)
        )
        (setq *listoflists-Col-Moments* list-of-lists-Col-Moments)
      ))


; Draw Handler attached to Rigid-Frame to draw the moments diagram
; Moments by Portal Method calculated elsewhere are used here.
(Define-handler  (Rigid-Frame Draw-Moments)
       (Window)
  (let* ((N-Bays (Slot-value self 'No-of-Bays))
    (Bay-Dim-List (slot-value Self 'Column-Spacings))
    (N-Storeys (Slot-value (Slot-value Self 'Floor-Outline-Instance)
          'no-of-floors))
    (Storey-Ht 4) ;;Tentative value (m)
    (Sxo 5)
    (Szo (+ 20 (* Storey-Ht N-Storeys)))
    (Big-List *listoflists-Col-Moments*))
    (Progn
      (Setq *CURRENT-WINDOW* Window)
;      (send-super-msg self Window)
      (Setq Sxi Sxo)
      (setq Szi Szo)
      (Rel-Move-X Sxi Szi -10)
      (Rel-Move-X Sxi Szi 60)
      (Dotimes (N N-Storeys )
       (Progn
  ;(print  *listoflists-Col-Moments*)
  (Dotimes (K  N-bays)
    (Let* ((Bay-i (First (nthcdr K Bay-Dim-List)))
     (Mom-Scale 0.1)
     (Moms-List
       (car (nthcdr K  Big-List)))
     (Last-col-mom (car (last Big-list)))
     ;(Mom 4)
     (I (- N-storeys N 1))
     (Mom (car (nthcdr I Moms-List))))
    ;(Print mom)
    (Rel-Move-Z Sxi Szi Storey-Ht) ;Draw Clmn-line left of
                                   ;Bay in Elevation
    (Draw-Moment Window Sxi Szi Bay-i Mom Mom-Scale Storey-Ht);Mmnt
    (setf Sxi (+ Sxi Bay-i))
  (Rel-Move-Z Sxi Szi Storey-Ht); Draw clmn-line to the
                                 ; right of last bay
  (setq Mom-last (car (nthcdr I Last-col-mom)))
  (Draw-Moment Window Sxi Szi Bay-i Mom-last

          Mom-Scale Storey-Ht);Mmnt
   ))
  (setq Sxi Sxo)
  (setq Szi (- Szi Storey-Ht))
```

```
    (Dotimes (I  N-Bays)
      (Let ((Bay-i (First (nthcdr I Bay-Dim-List))))
        (Rel-Move-X Sxi Szi Bay-i) ;Draw the Beam-line in Elevation
        (setq Sxi (+ Sxi Bay-i))))
      )
    (setq Sxi Sxo))
       (ghw:Draw-text window (floor (* 2 (- Sxi 4)))
           (floor (* 2 (+ 10 Szi (* N-Storeys Storey-Ht))))
          "Portal Method")
       (ghw:Draw-text window (floor (* 2 (- Sxi 4)))
           (floor (* 2 (+ 20 Szi (* N-Storeys Storey-Ht))))
          "Column-Moments"))))



(Defun Draw-Moment (Window Sxi Szi Bay-Size M Sc Ht)
                                  ;M - moment; Ht- Storey Ht; Sc-Scale
  (Let* ((Z Szi)
       (M2 (* Sc (/ M 2)))
       (X Sxi))
    (ghw:Draw-Line Window
   (floor (* 2 (- X M2))) (floor (* 2 Z)) (floor (* 2(+ X M2)))
      (floor (* 2 (- Z Ht)))
          :pen *red-pen* :units :square)))
```

Figure C-1 shows a screen displaying the moments in a 25 storey rigid frame subjected to a uniformly distributed lateral force. The portal method of analysis is used. The moments due to the vertical loads have to be added to these values to arrive at the design moments. The procedural routines listed so far in C-10 are used for generating these values.

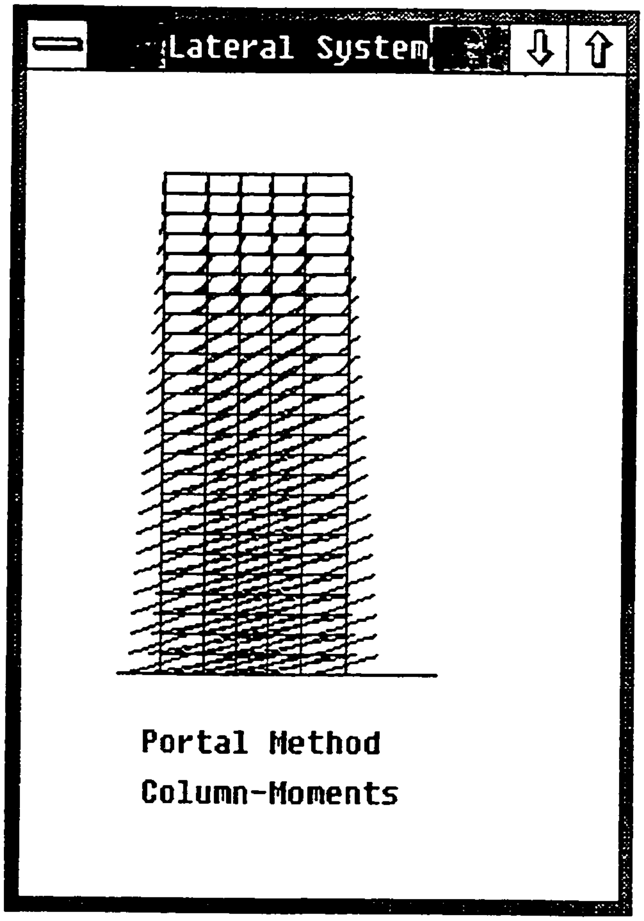


Figure C-1 Schematic of column moments in an arbitrary rigid frame, generated by Tall-D using portal method of analysis for a uniformly distributed lateral load.

APPENDIX D-1: Design Case 1 - Place du Canada

TABLE OF CONTENTS: Appendix D-1

TABLE OF FIGURES: APPENDIX D-1

Overall Configuration

Part-A: General Design Requirements

Name of the Building	PLACE-DU-CANADA
Length of Plot	70 m
Width of Plot	60 m
Finance Limit	$42 million
Max. no. of Floors	25
Required Floor Area	33405 m^2

Part-B: Initial Floor Outlines Considered

Outline	Length	Width	Perm-FAR	Status
FLOOR-1	65	55	12	OK-FAR
FLOOR-2	65	50	12	OK-FAR
FLOOR-3	65	40	12	OK-FAR
FLOOR-4	65	35	15	OK-FAR
FLOOR-5	65	30	15	OK-FAR
FLOOR-6	55	55	12	OK-FAR
FLOOR-7	55	50	12	OK-FAR
FLOOR-8	55	40	15	OK-FAR
FLOOR-9	55	35	15	OK-FAR
FLOOR-10	55	30	18	OK-FAR
FLOOR-11	45	40	15	OK-FAR
FLOOR-12	45	35	18	OK-FAR
FLOOR-13	45	30	18	OK-FAR
FLOOR-14	40	40	18	OK-FAR
FLOOR-15	40	35	18	OK-FAR
FLOOR-16	40	30	18	OK-FAR

Part-C: Status at the end of Level-I

Outline	Height(m)	BSR	#-Fl	Mch-Flrs	Status
FLOOR-1	48.0	0.9	12	1	OK-N-F
FLOOR-2	52.0	1.0	13	1	OK-N-F
FLOOR-3	64.0	1.6	16	1	OK-N-F
FLOOR-4	72.0	2.1	18	1	OK-N-F
FLOOR-5	84.0	2.8	21	1	OK-N-F
FLOOR-6	56.0	1.0	14	1	OK-N-F
FLOOR-7	60.0	1.2	15	1	OK-N-F
FLOOR-8	76.0	1.9	19	1	OK-N-F
FLOOR-9	84.0	2.4	21	1	OK-N-F
FLOOR-10	100.0	3.3	25	1	OK-N-F
FLOOR-11	92.0	2.3	23	1	OK-N-F
FLOOR-12	NIL	NIL	26	NIL	DEL-N-F
FLOOR-13	NIL	NIL	30	NIL	DEL-N-F
FLOOR-14	NIL	NIL	26	NIL	DEL-N-F
FLOOR-15	NIL	NIL	29	NIL	DEL-N-F
FLOOR-16	NIL	NIL	35	NIL	DEL-N-F

Part-D: Core Configuration

Outline	FL-L	FL-W	F-A-R	B-S-R	#-Flrs	SLNDR	Status	CORE	#-CORES	CORE-L	CORE-W	CORE-X	CORE-Y
FLOOR-1	65	55	12	0.9	12	MHS	OK-N-F	CENTRAL	1	25.2	21.3	19.9	16.8
FLOOR-2	65	50	12	1.0	13	MHS	OK-N-F	CENTRAL	1	25.2	19.4	19.9	15.3
FLOOR-3	65	40	12	1.6	16	MHS	OK-N-F	CENTRAL	1	25.2	15.5	19.9	12.3
FLOOR-4	65	35	15	2.1	18	MHS	OK-N-F	CENTRAL	1	25.2	13.6	19.9	10.7
FLOOR-5	65	30	15	2.8	21	MHS	OK-N-F	TWO	2	12.6	11.6	15.7	9.2
FLOOR-6	55	55	12	1.0	14	MHS	OK-N-F	CENTRAL	1	21.3	21.3	16.8	16.8
FLOOR-7	55	50	12	1.2	15	MHS	OK-N-F	CENTRAL	1	21.3	19.4	16.8	15.3
FLOOR-8	55	40	15	1.9	19	MHS	OK-N-F	CENTRAL	1	21.3	15.5	16.8	12.3
FLOOR-9	55	35	15	2.4	21	MHS	OK-N-F	CENTRAL	1	21.3	13.6	16.8	10.7
FLOOR-10	55	30	18	3.3	25	MHS	OK-N-F	CENTRAL	1	21.3	11.6	16.8	9.2
FLOOR-11	45	40	15	2.3	23	MHS	OK-N-F	CENTRAL	1	17.4	15.5	13.8	12.3

Part-E: Evaluation Preferences Supplied by the Designer

Flexibility of rentable areas	100
Window line for rentable areas	100
Suitability for lateral structural system	50
High rentability at ground level	0
Travel distance from core to window-line	100
Clarity of circulation of rental areas	50
Daylight and view for core areas	0
Service connections at roof	0
Service connections at ground	0
General Energy Efficiency	100

Part-F: Floor Evaluation Result

(Not Sorted by Rank)

Floor Plan	#Floors	Core-Type	Eval-Value	Rank
FLOOR-1	12	CENTRAL	2050.0	2
FLOOR-2	13	CENTRAL	2050.0	2
FLOOR-3	16	CENTRAL	1950.0	3
FLOOR-4	18	CENTRAL	1950.0	3
FLOOR-5	21	TWO	1650.0	4
FLOOR-6	14	CENTRAL	2150.0	1
FLOOR-7	15	CENTRAL	2050.0	2
FLOOR-8	19	CENTRAL	2050.0	2
FLOOR-9	21	CENTRAL	1950.0	3
FLOOR-10	25	CENTRAL	1950.0	3
FLOOR-11	23	CENTRAL	2050.0	2

SUMMARY OF STRUCTURAL LAYOUT ALTERNATIVES

(Note: Sample alternatives have also been presented graphically at the end)

Alternative Structural Schemes for Layout#8

Alternative Structural Scheme for: Layout#8

Layout#8: Concrete-Rfr-Scheme-9

Structure Type: Concrete Rigid Frame Scheme
Major Stability Element: Rigid-Frame
Column LayoutType: Grid-2d
Approximately equal column grid.
Alternative Grid Spacings:
(Not in a particular order)
Note: Only some of the frames used for rigid-frame action.
i.e. Not all frames have be moment connected.
Alternative Grid Spacings:

Bay: 7.9m
Aisle: 8.0m
Bay: 11.0m
Aisle: 13.3m
7.9x8.0m Grid with 7 Bays & 5 Aisles
11.0x13.3m Grid with 5 Bays & 3 Aisles
Grid with unequal column spacing
Alternative Grid Spacings:
Bay: 6.1m Aisle: 13.3m
Bay: 4.6m Aisle: 13.3m
Column LayoutType: Grid-Aligned-To-Core
Clear Span between core and window line:
Front & Back edges to Core: 12.3m
Two Spans between core and window line:
Left & Right sides - Each span of: NILm
1.Column Spacing:
Perimeter (Front & Back):4.6m
Perimeter (Sides):5.7m
Structural Grid inside the Core area: 5.3m x 5.2m

Alternative Structural Scheme for: Layout#8

Layout#8: Concrete-Rfr-Scheme-10

Structure Type: Concrete Rigid Frame Scheme
Major Stability Element: Perimeter-Frame
Column LayoutType: Perimete?r-Based
Clear Span between core and window line:
Front & Back edges to Core: 12.3m
Two Spans between core and window line:
Left & Right sides - Each span of: NILm
1.Column Spacing:
Perimeter (Front & Back):4.6m
Perimeter (Sides):5.7m
2.Column Spacing:
Perimeter (Front & Back):3.1m
Perimeter (Sides):3.1m
Structural Grid inside the Core area: 5.3m x 5.2m

Alternative Structural Scheme for: Layout#8

Layout#8: Steel-Scheme-7

Structure Type: Steel Rigid Frame Scheme
Major Stability Element: Rigid-Frame
Column LayoutType: Grid-2d
Approximately equal column grid.
Alternative Grid Spacings:
(Not in a particular order)
Note: Only some of the frames used for rigid-frame action.
i.e. Not all frames have be moment connected.
Alternative Grid Spacings:
Bay: 11.0m
Aisle: 13.3m
Bay: 7.9m
Aisle: 8.0m
11.0x13.3m Grid with 5 Bays & 3 Aisles

```
    7.9x8.0m Grid with 7 Bays & 5 Aisles
 Grid with unequal column spacing
   Alternative Grid Spacings:
   Bay: 6.1m  Aisle: 13.3m
   Bay: 4.6m  Aisle: 13.3m
 Column LayoutType: Grid-Aligned-To-Core
   Clear Span between core and window line:
       Front & Back edges to Core: 12.3m
   Two Spans between core and window line:
    Left & Right sides - Each span of: NILm
  1.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.7m
  2.Column Spacing:
     Perimeter (Front & Back):4.6m
     Perimeter (Sides):4.4m
   Structural Grid inside the Core area: 5.3m x 5.2m
```

```
Alternative Structural Scheme for: Layout#8
```

Layout#8: Steel-Scheme-8

```
 Structure Type: Steel Rigid Frame Scheme
 Major Stability Element: Perimeter-Frame
 Column LayoutType: Perimeter-Based
   Clear Span between core and window line:
       Front & Back edges to Core: 12.3m
   Two Spans between core and window line:
    Left & Right sides - Each span of: NILm
  1.Column Spacing:
     Perimeter (Front & Back):3.1m
     Perimeter (Sides):3.1m
  2.Column Spacing:
     Perimeter (Front & Back):4.6m
     Perimeter (Sides):5.7m
   Structural Grid inside the Core area: 3.0m x 3.1m
```

```
Alternative Structural Scheme for: Layout#8
```

Layout#8: Braced-Perimeter-5

```
 Structure Type: Braced Frame Scheme
 Major Stability Element: Braced-Frame-Perimeter
 Column LayoutType: Perimeter-Based
   Clear Span between core and window line:
       Front & Back edges to Core: 12.3m
   Two Spans between core and window line:
    Left & Right sides - Each span of: NILm
  1.Column Spacing:
     Perimeter (Front & Back):9.2m
     Perimeter (Sides):10.0m
  2.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.7m
   Structural Grid inside the Core area: 5.3m x 5.2m
```

```
Alternative Structural Scheme for: Layout#8
```

Layout#8: Braced-Internal-6

```
 Structure Type: Braced Frame Scheme
 Major Stability Element: Braced-Frame-Internal
 Column LayoutType: Grid-2d
  Approximately equal column grid.
  Alternative Grid Spacings:
  (Not in a particular order)
  Note: Only some of the frames used for rigid-frame action.
       i.e. Not all frames have be moment connected.
   Alternative Grid Spacings:
   Bay:   11.0m
   Aisle: 13.3m
   Bay:   7.9m
   Aisle: 8.0m
    11.0x13.3m Grid with 5 Bays & 3 Aisles
    7.9x8.0m Grid with 7 Bays & 5 Aisles
 Grid with unequal column spacing
   Alternative Grid Spacings:
   Bay: 6.1m  Aisle: 13.3m
 Column LayoutType: Grid-Aligned-To-Core
   Clear Span between core and window line:
       Front & Back edges to Core: 12.3m
   Two Spans between core and window line:
    Left & Right sides - Each span of: NILm
  1.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.7m
   Structural Grid inside the Core area: 5.3m x 5.2m
```

Approximate Column Sizes: Layout#8

Structural Scheme: Concrete-Rfr-Scheme-9

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 4.6m Aisles: 13.3m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
19-19	300	300	300	300
18-16	300	300	300	300
15-13	300	300	300	300
12-10	300	300	400	300
9- 7	300	300	550	300
6- 4	400	400	650	400
3- 1	550	550	750	550

Column concrete volume for the above alternative:950.01 cu.m.

Alternative with internal grid of Bays: 6.1m Aisles: 13.3m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
19-19	300	300	300	300
18-16	300	300	300	300
15-13	300	300	400	300
12-10	300	300	550	300
9- 7	400	400	650	400
6- 4	550	550	750	550
3- 1	650	650	850	650

Column concrete volume for the above alternative:1000.44cu.m.

Alternative with internal grid of Bays: 11.0m Aisles: 13.3m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
19-19	300	300	300	300
18-16	300	300	300	300
15-13	300	300	550	300
12-10	550	550	750	550
9- 7	650	650	850	650
6- 4	750	750	950	750
3- 1	850	850	1000	850

Column concrete volume for the above alternative:988.92 cu.m.

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX: 4.6m YY: 4.4m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	300	300	300
9- 7	300	300	300
6- 4	400	400	400
3- 1	550	550	550

Column concrete volume for the above alternative: 482. cu.m.

Structural Scheme: Concrete-Rfr-Scheme-10

Column Layout Type: Perimeter-Based

Details of Columns :

Alternative with Perimeter column spacings XX: 4.6m YY: 5.7m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	300	300	300
9- 7	300	300	300
6- 4	400	400	400
3- 1	550	550	550

Column concrete volume for the above alternative: 482. cu.m.

--**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*

Alternative with Perimeter column spacings XX: 3.1m YY: 3.1m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	300	300	300
9- 7	300	300	300
6- 4	300	300	300
3- 1	400	400	400

Column concrete volume for the above alternative: 487. cu.m.

Structural Scheme: Steel-Scheme-7

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 4.6m Aisles: 13.3m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
19-19	(W150 150)	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W310 310)	(W200 200)
12-10	(W310 310)	(W310 310)	(W310 310)	(W200 200)
9- 7	(W310 310)	(W310 310)	(W310 310)	(W250 250)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for this alternative: 231000.0 KG

Alternative with internal grid of Bays: 6.1m Aisles: 13.3m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
19-19	(W150 150)	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W250 250)	(W200 200)
15-13	(W250 250)	(W250 250)	(W310 310)	(W200 200)
12-10	(W310 310)	(W310 310)	(W310 310)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Column Steel weight for this alternative: 180600.0 KG

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX:4.6m YY:4.4m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)

9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 184800.0 KG

Alternative with Perimeter column spacings XX: 6.1m YY: 5.7m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 155400.0 KG

Structural Scheme: Steel-Scheme-8

Column Layout Type: Perimeter-Based

Details of Columns :

Alternative with Perimeter column spacings XX: 4.6m YY: 4.4m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 184800.0 KG

Alternative with Perimeter column spacings XX: 3.1m YY:3.1m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W200 200)	(W200 200)	(W200 200)
12-10	(W250 250)	(W250 250)	(W250 250)

12-10	(W250 250)	(W250 250)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 243600.0 KG

--**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*

Structural Scheme: Braced-Internal-6

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 6.1m Aisles: 13.3m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
19-19	(W150 150)	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W250 250)	(W200 200)
15-13	(W250 250)	(W250 250)	(W310 310)	(W200 200)
12-10	(W310 310)	(W310 310)	(W310 310)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Column Steel weight for this alternative: 180600.0 KG

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX: 6.1m YY: 5.7m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
19-19	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 155400.0 KG

--**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*-**-*

Table D1: Summary of gravity system alternatives for Layout#8.

Construction Material	Type of Gravity System	Floor Depth OR Beam/slab size mm x mm	Relative Preference	Column Layout Type	Floor Depth OR Beam/slab size mm x mm	Relative Preference	Column Layout Type
Concrete		Structural Scheme: Concrete-Rfr-Scheme-9			Structural Scheme: Concrete-Rfr-Scheme-10		
	One-Way Beam-Slab	1400 deep 210 slab	3	Grid-Aligned-To-Core	1400 deep	3	Perimeter-Based
	Joist Slab	1250deep 270 joists	1	Grid-Aligned-To-Core	1250deep 270 joists	1	Perimeter-Based
Floor	Waffle Slab	300 slab	2	Grid-Aligned-To-Core	300 slab	2	Perimeter-Based
	Band-Beam Slab	1250 band 270 slab	4	Grid-Aligned-To-Core	1150 band 170 slab	4	Perimeter-Based
	Hollow-Core Slab	420 slab	5	Grid-Aligned-To-Core	420 slab	5	Perimeter-Based
Systems	One-Way Beam-Slab	1400 deep	3	Grid-2D			
	Joist Slab	950 deep 250 joists	1	Grid-2D			
	Waffle Slab		2	Grid-2D			
	Band-Beam Slab	1150 deep 170 slab	4	Grid-2D			

...continued

Table D1: Summary of gravity system alternatives for Layout#8 (continued).

		Structural Scheme: Steel-Scheme-7			Structural Scheme: Braced-Internal-6		
Composite Steel Deck Floor Systems with Steel Beams	Tapered Beam	250x510	3	Grid-2D	250x510	3	Grid-2D
	Truss Beam	610deep +60 deck	1	Grid-2D	610deep +60 deck	1	Grid-2D
	Haunch Beam	250x410	2	Grid-2D	250x410	2	Grid-2D
	Parallel Beam	250x500	4	Grid-2D	250x500	4	Grid-2D
	Stub Girder	250x510	5	Grid-2D	250x10	5	Grid-2D
	Tapered Beam	310x630	3	Grid-Aligned-To-Core	310x630	3	Grid-Aligned-To-Core
	Truss Beam	750 deep +70 deck	1	Grid-Aligned-To-Core	750 deep +70 deck	1	Grid-Aligned-To-Core
	Haunch Beam	310x500	2	Grid-Aligned-To-Core	310x500	2	Grid-Aligned-To-Core
	Parallel Beam	200x410	4	Grid-Aligned-To-Core	200x410	4	Grid-Aligned-To-Core
	Stub Girder	310x630	5	Grid-Aligned-To-Core	310x630	5	Grid-Aligned-To-Core
		Structural Scheme: Steel-Scheme-8					
	Tapered Beam	310x630	3	Perimeter-Based			
	Truss Beam	750 deep +70 deck	1	Perimeter-Based			
	Haunch Beam	310x500	2	Perimeter-Based			
	Parallel Beam	250x500	4	Perimeter-Based			
	Stub Girder	310x630	5	Perimeter-Based			

Note: 1 is most preferred; Based on historical in-place cost, on a relative basis.

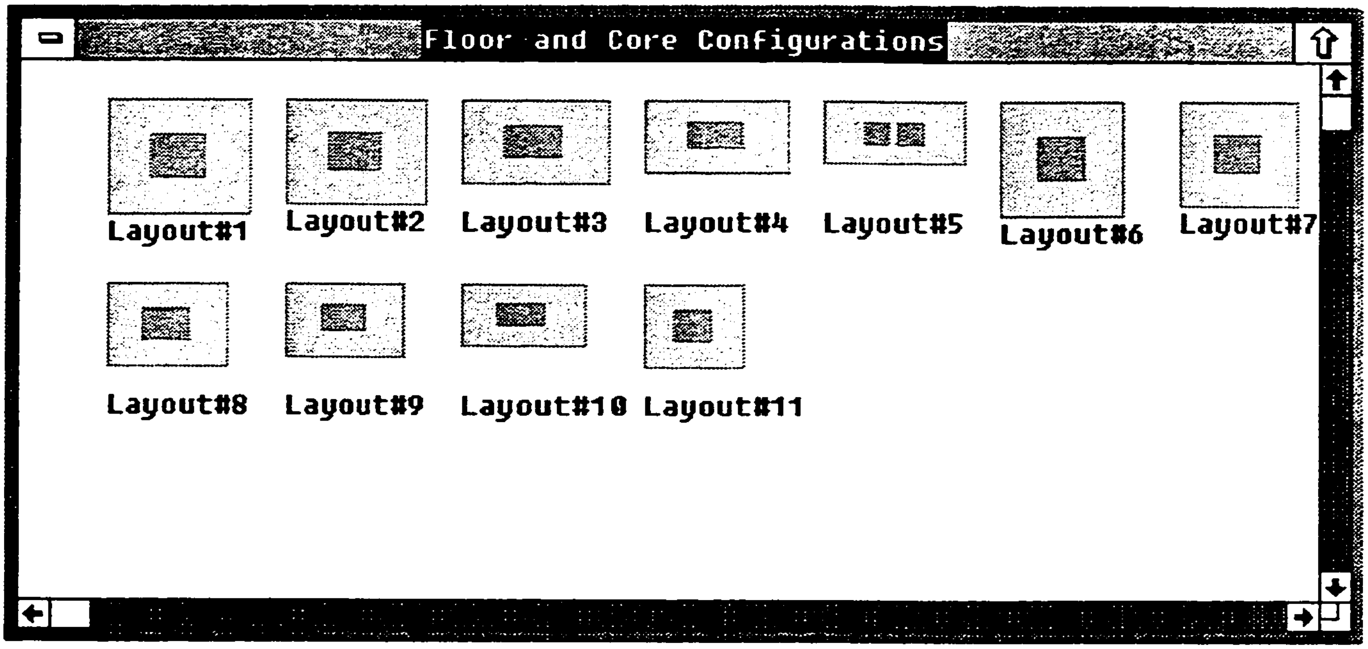
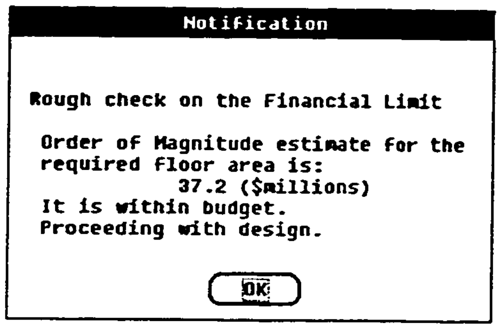


Fig. D1- 1 Floor layout alternatives retained at end of Phase-1 of design.

(a)



(b)

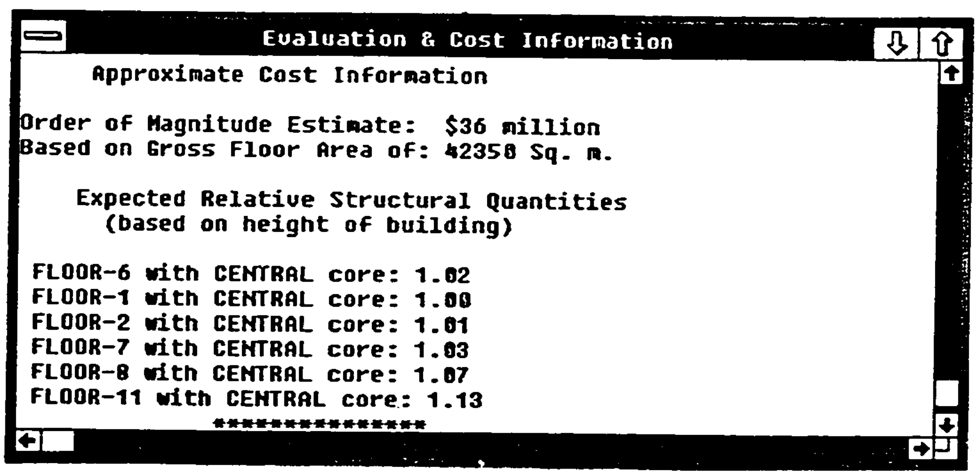


Fig. D1- 2 (a) Initial check on cost

(b) Revised cost estimate after assigned number of floors.

Sample Images of Alternative for Layout#8

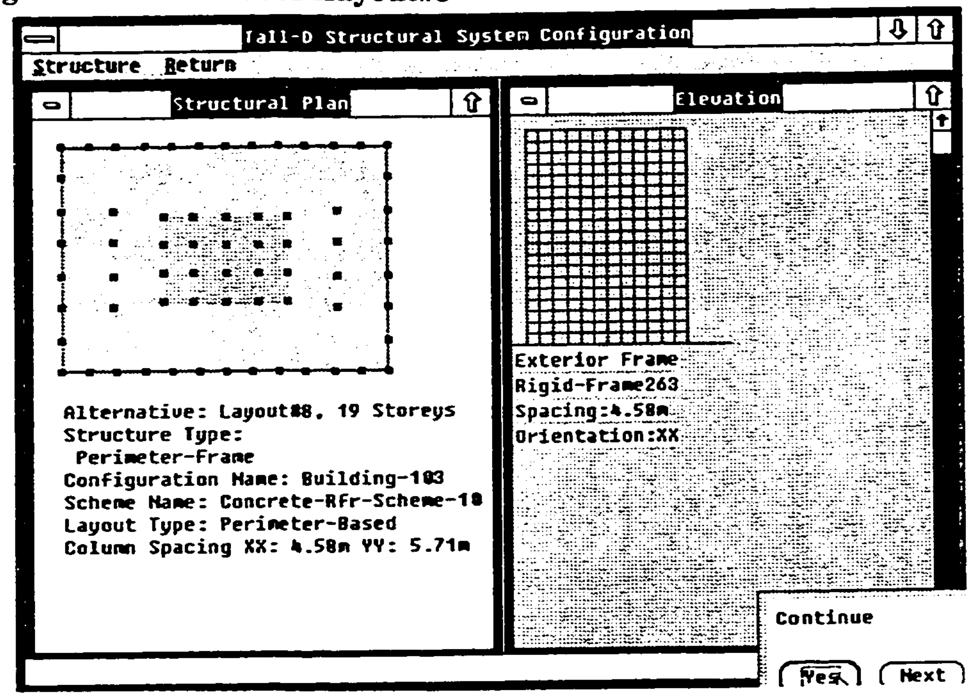


Fig. D1- 3 Perimeter-based rigid frame strucural system alternative for Layout#8: Perimeter column spacing 4.58m and 5.71m

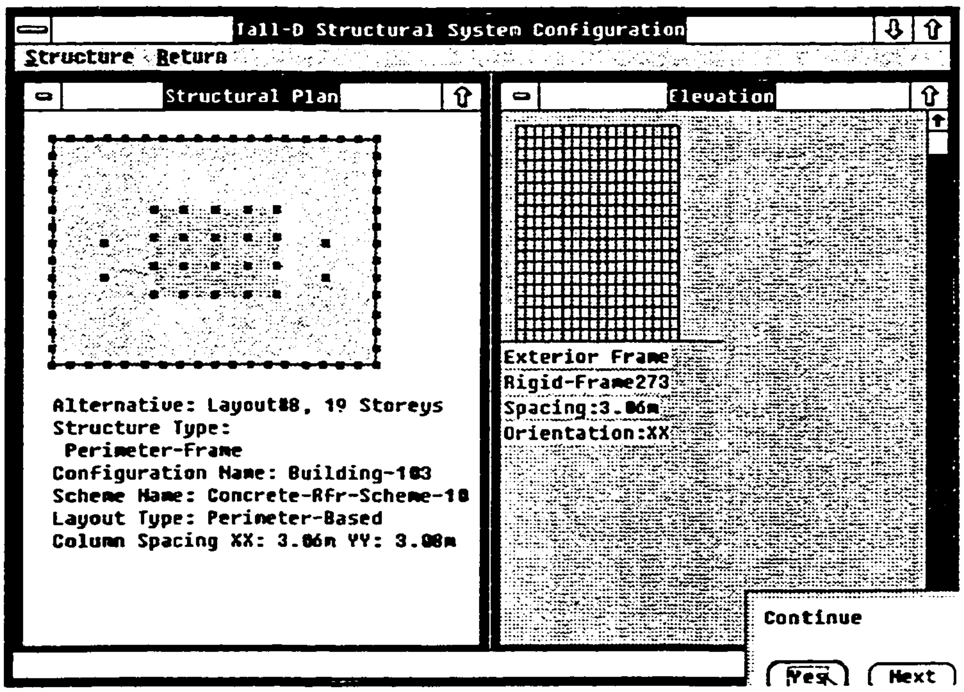


Fig. D1- 4 Perimeter-based rigid frame strucural system alternative for Layout#8: Perimeter column spacing 3.06m and 3.08m

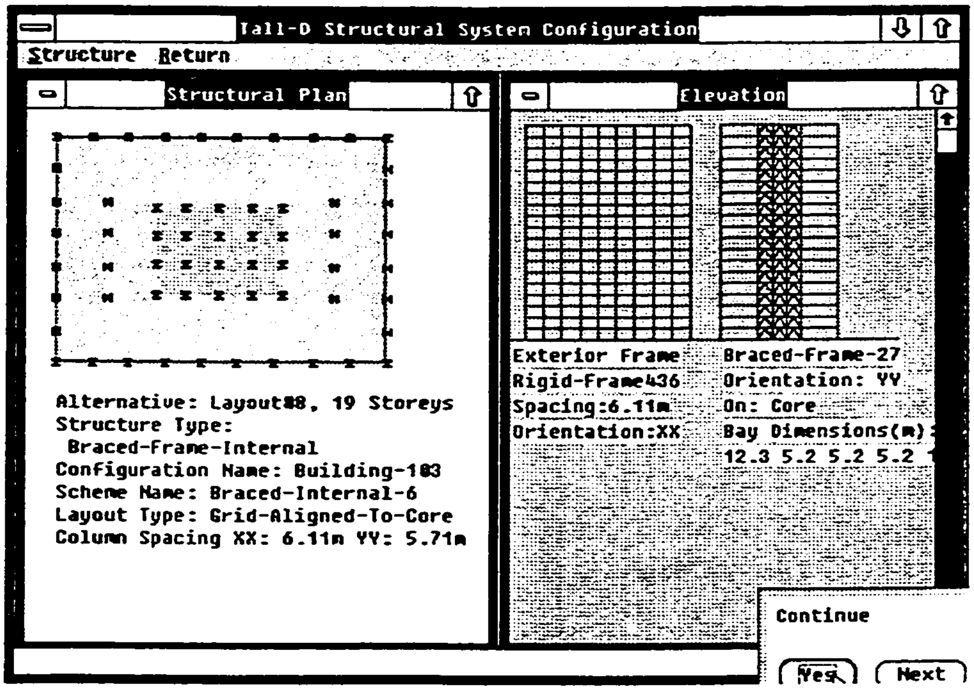


Fig. D1- 5 Braced frame strucural system alternative for Layout#8: Perimeter column spacing 6.11m and 5.71m

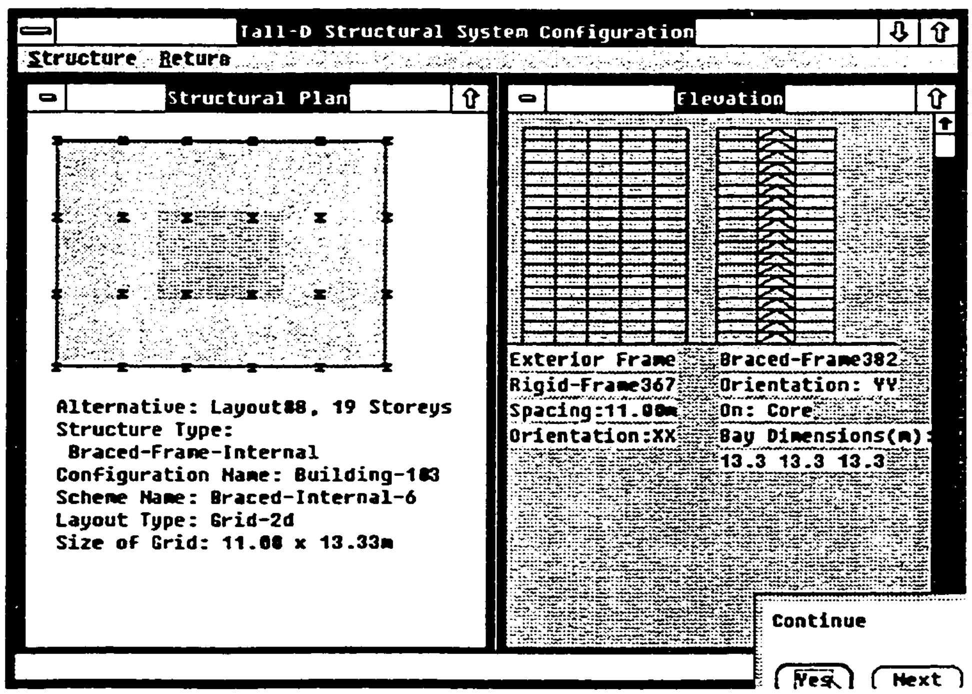


Fig. D1- 6 Braced frame strucural system alternative for Layout#8: Perimeter column spacing 11.0m and 13.3m.

Alternative Structural Schemes for Layout#9

```
Alternative Structural Scheme for: Layout#9
```

Layout#9: Fr-Shearwl-Scheme-9

```
 Structure Type: Frame ShearWall Scheme
 Major Stability Element: Frame-Shearwall
 Column LayoutType: As-Per-Shearwalls
   Clear Span between core and window line:
       Front & Back edges to Core: 10.7m
   Clear Span between core and window line:
       Left & Right edges to Core: 16.8m
  Note: 2 span(s) of Girders at Left and
                   Right Ends each of 8.4m
  1.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.8m
  2.Column Spacing:
     Perimeter (Front & Back):6.9m
     Perimeter (Sides):7.0m
  3.Column Spacing:
     Perimeter (Front & Back):4.6m
     Perimeter (Sides):4.4m
   Number of Shear Walls in Core:
     YY direction (North-South): 3
     XX direction (East-West):   2
        (Note: XX walls are the flanges of YY walls)
```

```
Alternative Structural Scheme for: Layout#9
```

Layout#9: Steel-Scheme-7

```
 Structure Type: Steel Rigid Frame Scheme
 Major Stability Element: Rigid-Frame
 Column LayoutType: Grid-2d
 Grid with unequal column spacing
   Alternative Grid Spacings:
   Bay: 4.6m  Aisle: 11.7m
 Column LayoutType: Grid-Aligned-To-Core
   Clear Span between core and window line:
       Front & Back edges to Core: 10.7m
   Clear Span between core and window line:
       Left & Right edges to Core: 16.8m
   Note: 2 span(s) of Girders at Left and
                   Right Ends each of 8.4m
  1.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.8m
  2.Column Spacing:
     Perimeter (Front & Back):4.6m
     Perimeter (Sides):4.4m
   Structural Grid inside the Core area: 5.3m x 6.8m
```

```
Alternative Structural Scheme for: Layout#9
```

Layout#9: Steel-Scheme-8

```
 Structure Type: Steel Rigid Frame Scheme
 Major Stability Element: Perimeter-Frame
 Column LayoutType: Perimeter-Based
   Clear Span between core and window line:
       Front & Back edges to Core: 10.7m
   Clear Span between core and window line:
       Left & Right edges to Core: 16.8m
   Note: 2 span(s) of Girders at Left and
                   Right Ends each of 8.4m
  1.Column Spacing:
     Perimeter (Front & Back):3.1m
     Perimeter (Sides):2.9m
  2.Column Spacing:
     Perimeter (Front & Back):4.6m
     Perimeter (Sides):4.4m
   Structural Grid inside the Core area: 5.3m x 6.8m
```

```
Alternative Structural Scheme for: Layout#9
```

Layout#9: Braced-Perimeter-5

```
 Structure Type: Braced Frame Scheme
 Major Stability Element: Braced-Frame-Perimeter
 Column LayoutType: Perimeter-Based
   Clear Span between core and window line:
       Front & Back edges to Core: 10.7m
   Clear Span between core and window line:
       Left & Right edges to Core: 16.8m
  Note: 2 span(s) of Girders at Left and
                   Right Ends each of 8.4m
  1.Column Spacing:
     Perimeter (Front & Back):9.2m
     Perimeter (Sides):8.7m
  2.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.8m
   Structural Grid inside the Core area: 5.3m x 6.8m
```

```
Alternative Structural Scheme for: Layout#9
```

Layout#9: Braced-Internal-6

```
 Structure Type: Braced Frame Scheme
 Major Stability Element: Braced-Frame-Internal
 Column LayoutType: Grid-2d
 Grid with unequal column spacing
   Alternative Grid Spacings:
   Bay: 6.1m  Aisle: 11.7m
 Column LayoutType: Grid-Aligned-To-Core
   Clear Span between core and window line:
```

```
      Front & Back edges to Core: 10.7m
   Clear Span between core and window line:
      Left & Right edges to Core: 16.8m
  Note: 2 span(s) of Girders at Left and
                 Right Ends each of 8.4m
  1.Column Spacing:
     Perimeter (Front & Back):6.1m
     Perimeter (Sides):5.8m
   Structural Grid inside the Core area: 5.3m x 6.8m
```

Approximate Column Sizes: Layout#9

Structural Scheme: Fr-Shearwl-Scheme-9

Column Layout Type: As-Per-Shearwalls

Shearwall Details:

```
Number of Shearwalls (with flanges) in Core: 3
Concrete strength: 40MPa
Spacing of Shearwalls:10.7m
1. Thickness of wall at base: 375mm
    Wall Shape: C
2. Thickness of wall at base: 375mm
    Wall Shape: I
3. Thickness of wall at base: 375mm
    Wall Shape: C-INVERTED
Shearwall concrete volume for this alternative:  1347. cu.m.
  All Walls may taper to 250mm at top;
  May be terminated in some cases;
  & also in cases where elevator banks drop off.
```

Details of Columns:

Alternative with Perimeter column spacings XX: 6.1m YY:5.8m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	300	300	300
9- 7	400	400	400
6- 4	550	550	550
3- 1	550	550	550

Column and Shearwall concrete volume for this alternative:
1860. cu.m.

Alternative with Perimeter column spacings XX: 6.9m YY:7.0m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	400	400	400
9- 7	550	550	550
6- 4	550	550	550
3- 1	650	650	650

Column and Shearwall concrete volume for this alternative:
1959. cu.m.

Alternative with Perimeter column spacings XX:4.6m YY:4.4m
CONCRETE Strength: 40MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	300	300	300
18-16	300	300	300
15-13	300	300	300
12-10	300	300	300
9- 7	300	300	300
6- 4	400	400	400
3- 1	550	550	550

Column and Shearwall concrete volume for this alternative:
1840. cu.m.

Structural Scheme: Steel-Scheme-7

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 4.6m Aisles: 11.7m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
21-19	(W200 200)	(W200 200)	(W200 200)	(W150 150)
18-16	(W200 200)	(W200 200)	(W310 310)	(W200 200)
15-13	(W250 250)	(W250 250)	(W310 310)	(W200 200)
12-10	(W310 310)	(W310 310)	(W310 310)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for this alternative: 231000.0 KG

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX: 4.6m YY:4.4m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 176400.0 KG

Alternative with Perimeter column spacings XX: 6.1m YY:5.8m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 147000.0 KG

Structural Scheme: Steel-Scheme-8

Column Layout Type: Perimeter-Based

Details of Columns :

Alternative with Perimeter column spacings XX: 4.6m YY: 4.4m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W250 250)	(W250 250)	(W250 250)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 176400.0 KG

Alternative with Perimeter column spacings XX: 3.1mYY: 2.9m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	(W150 150)	(W150 150)	(W150 150)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W200 200)	(W200 200)	(W200 200)
12-10	(W250 250)	(W250 250)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 235200.0 KG

Structural Scheme: Braced-Internal-6

Column Layout Type: Grid-2d

Details of Columns:

Alternative with internal grid of Bays: 6.1m Aisles: 11.7m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)	Interior (mm)
21-19	(W200 200)	(W200 200)	(W200 200)	(W150 150)
18-16	(W250 250)	(W250 250)	(W310 310)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)	(W200 200)
12-10	(W310 310)	(W310 310)	(W310 310)	(W250 250)
9- 7	(W310 310)	(W310 310)	(W360 360)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W360 360)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W360 360)	(W310 310)

Column Steel weight for this alternative: 180600.0 KG

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX: 6.1m YY:5.8m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
21-19	(W200 200)	(W200 200)	(W200 200)
18-16	(W200 200)	(W200 200)	(W200 200)
15-13	(W310 310)	(W310 310)	(W310 310)
12-10	(W310 310)	(W310 310)	(W310 310)
9- 7	(W310 310)	(W310 310)	(W310 310)
6- 4	(W310 310)	(W310 310)	(W310 310)
3- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for the above alternative: 147000.0 KG

Sample Images of Alternative for Layout#9

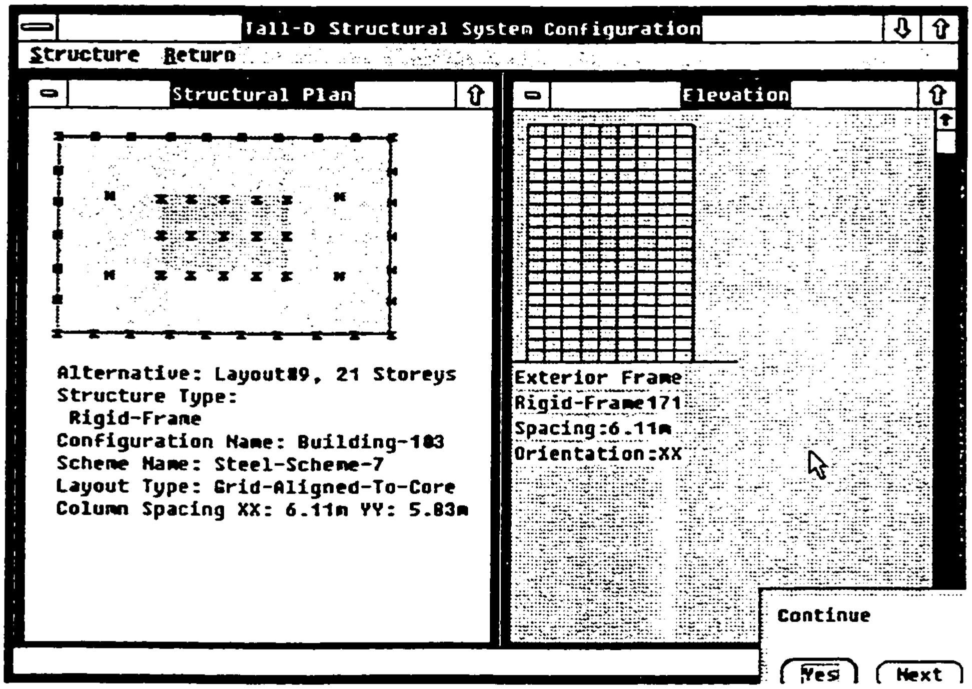


Fig. D1- 7 Rigid frame strucural system alternative for Layout#9: Perimeter column spacing 6.11m and 5.83m

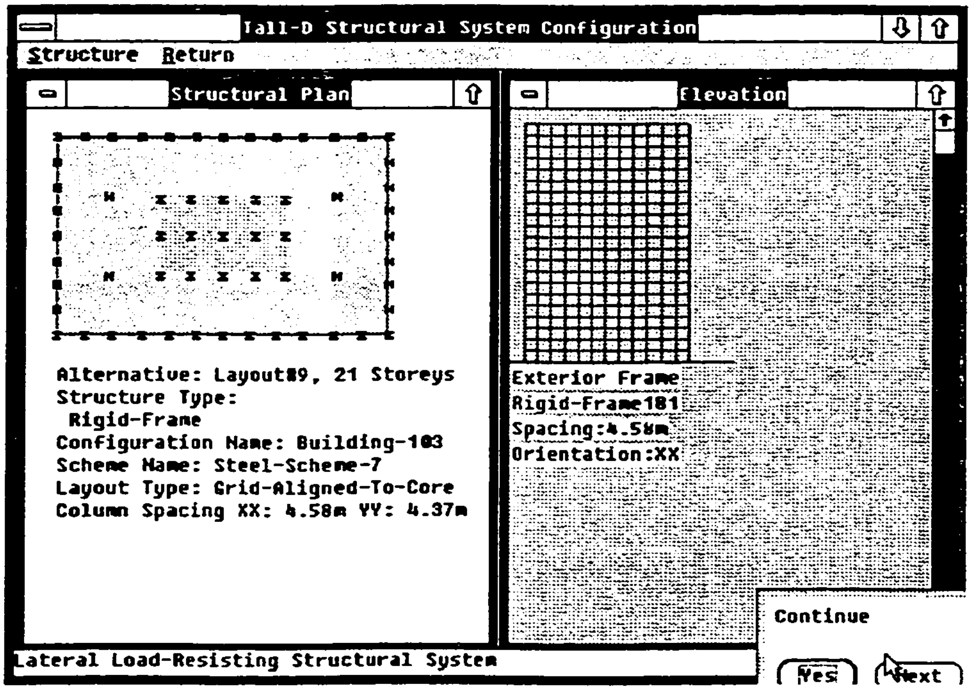


Fig. D1- 8 Rigid frame strucural system alternative for Layout#9: Perimeter column spacing 4.58m and 4.37m

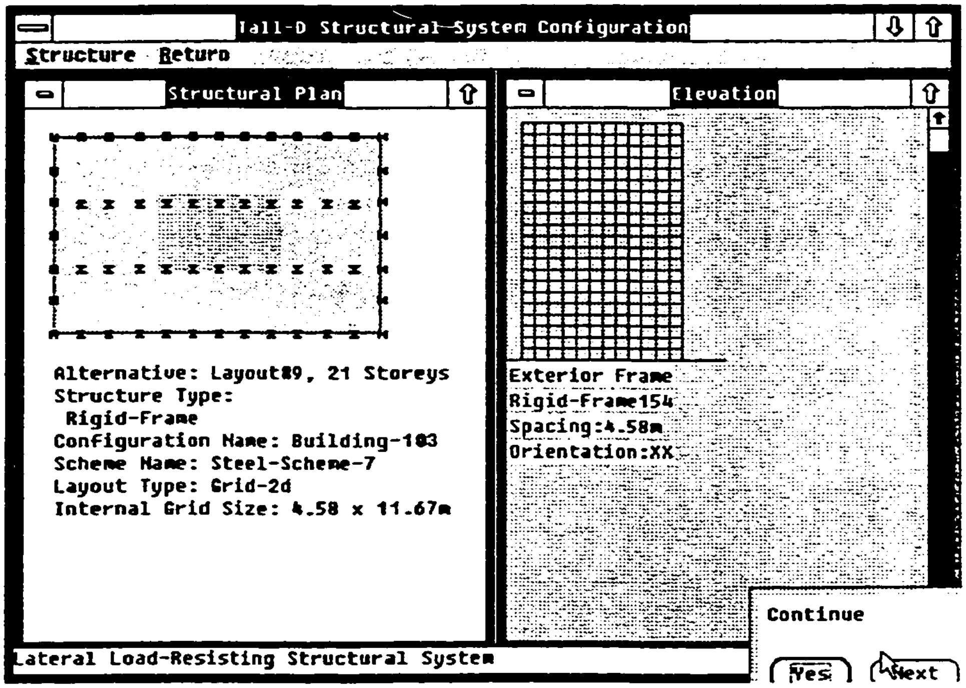


Fig. D1- 9 Rigid frame strucural system alternative for Layout#9: Column spacing - Aisles: 4.58m, Bays:11.67m.

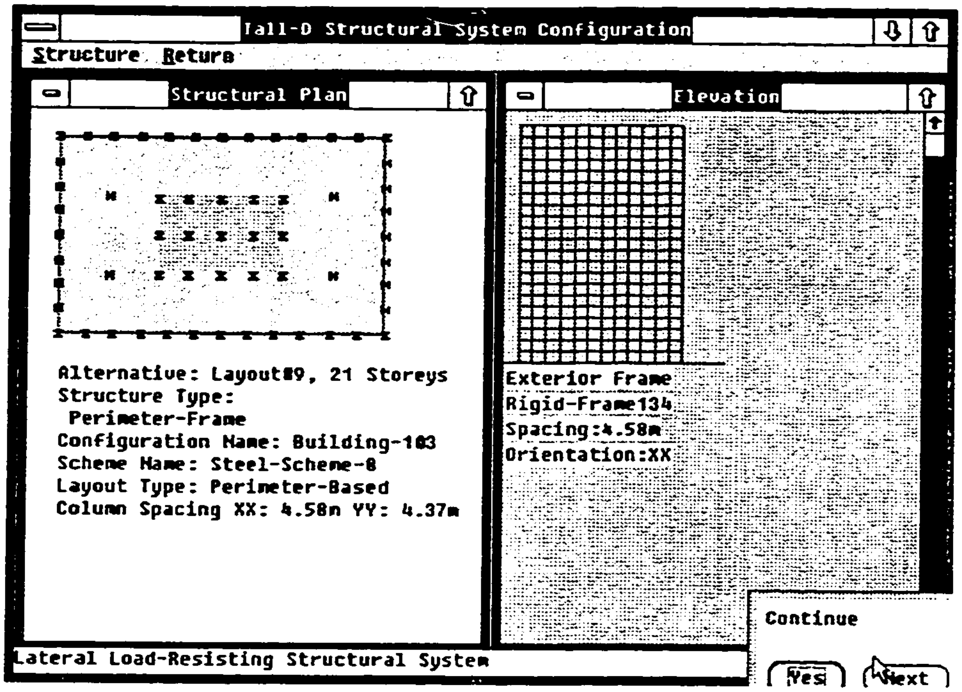


Fig. D1- 10 Perimeter based rigid frame strucural system alternative for Layout#9: Perimeter column spacing 4.58m and 4.37m

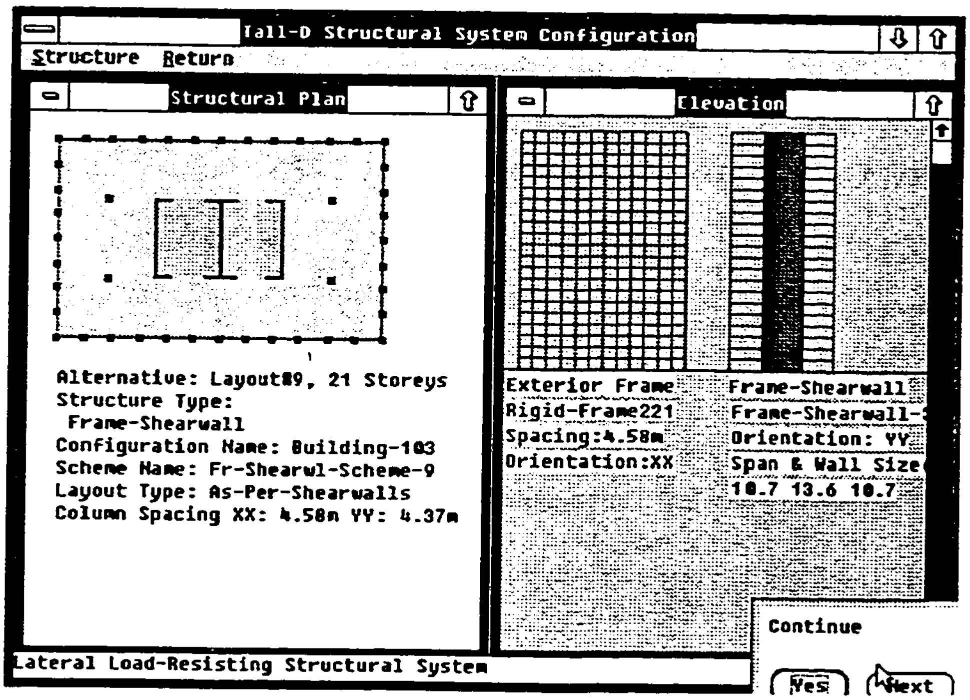


Fig. D1- 11 Frame-shearwall interaction strucural system alternative for Layout#9: Perimeter column spacing 4.58m and 4.37m

APPENDIX D-2: Design Case 2 - IBM-Marathon

TABLE OF CONTENTS: Appendix D-2

TABLE OF FIGURES: Appendix D-2

Overall Configuration

Part-A: General Design Requirements

Name of the Building	IBM4-MARATHON
Length of Plot	130 m
Width of Plot	70 m
Finance Limit	$ 200 million
Max. no. of Floors	50
Required Floor Area	65790 m^2

Part-B: Initial Floor Outlines Considered

Outline	Length	Width	Perm-FAR	Status
FLOOR-1	85	65	15	OK-FAR
FLOOR-2	85	55	15	OK-FAR
FLOOR-3	85	50	15	OK-FAR
FLOOR-4	85	40	18	OK-FAR
FLOOR-5	85	35	18	OK-FAR
FLOOR-6	85	30	18	OK-FAR
FLOOR-7	75	65	15	OK-FAR
FLOOR-8	75	55	15	OK-FAR
FLOOR-9	75	50	15	OK-FAR
FLOOR-10	75	40	18	OK-FAR
FLOOR-11	75	35	18	OK-FAR
FLOOR-12	75	30	18	OK-FAR
FLOOR-13	65	65	15	OK-FAR
FLOOR-14	65	55	18	OK-FAR
FLOOR-15	65	50	18	OK-FAR
FLOOR-16	65	40	18	OK-FAR
FLOOR-17	65	35	18	OK-FAR
FLOOR-18	65	30	18	OK-FAR
FLOOR-19	55	55	18	OK-FAR
FLOOR-20	55	50	18	OK-FAR
FLOOR-21	55	40	18	OK-FAR
FLOOR-22	55	35	18	OK-FAR
FLOOR-23	55	30	18	OK-FAR
FLOOR-24	45	40	18	OK-FAR
FLOOR-25	45	35	18	OK-FAR
FLOOR-26	45	30	18	OK-FAR
FLOOR-27	40	40	18	OK-FAR
FLOOR-28	40	35	18	OK-FAR
FLOOR-29	40	30	18	OK-FAR

Part-C: Status at the end of Level-I

Outline	Height(m)	BSR	#-Fl	Mch-Flrs	Status
FLOOR-1	64.0	1.0	16	1	OK-N-F
FLOOR-2	76.0	1.4	19	1	OK-N-F
FLOOR-3	80.0	1.6	20	1	OK-N-F
FLOOR-4	100.0	2.5	25	1	OK-N-F
FLOOR-5	116.0	3.3	29	1	OK-N-F
FLOOR-6	136.0	4.5	34	2	OK-N-F
FLOOR-7	72.0	1.1	18	1	OK-N-F
FLOOR-8	84.0	1.5	21	1	OK-N-F
FLOOR-9	92.0	1.8	23	1	OK-N-F
FLOOR-10	112.0	2.8	28	1	OK-N-F
FLOOR-11	132.0	3.8	33	2	OK-N-F
FLOOR-12	156.0	5.2	39	2	OK-N-F
FLOOR-13	80.0	1.2	20	1	OK-N-F
FLOOR-14	96.0	1.7	24	1	OK-N-F
FLOOR-15	104.0	2.1	26	1	OK-N-F
FLOOR-16	136.0	3.4	34	2	OK-N-F
FLOOR-17	152.0	4.3	38	2	OK-N-F
FLOOR-18	176.0	5.9	44	2	OK-N-F
FLOOR-19	112.0	2.0	28	1	OK-N-F
FLOOR-20	124.0	2.5	31	1	OK-N-F
FLOOR-21	156.0	3.9	39	2	OK-N-F
FLOOR-22	180.0	5.1	45	2	OK-N-F
FLOOR-23	NIL	NIL	52	NIL	DEL-N-F
FLOOR-24	192.0	4.8	48	2	OK-N-F
FLOOR-25	NIL	NIL	54	NIL	DEL-N-F
FLOOR-26	NIL	NIL	64	NIL	DEL-N-F
FLOOR-27	NIL	NIL	53	NIL	DEL-N-F
FLOOR-28	NIL	NIL	61	NIL	DEL-N-F
FLOOR-29	NIL	NIL	72	NIL	DEL-N-F

Part-D: Core Configuration

Outline	FL-L	FL-W	F-A-R	B-S-R	#-Flrs	SLNDR	Status	CORE	#-CORES	CORE-L	CORE-W	CORE-X	CORE-Y
FLOOR-1	85	65	15	1.0	16	MHS	OK-N-F	CENTRAL	1	38.0	29.1	23.5	18.0
FLOOR-2	85	55	15	1.4	19	MHS	OK-N-F	CENTRAL	1	38.0	24.6	23.5	15.2
FLOOR-3	85	50	15	1.6	20	MHS	OK-N-F	CENTRAL	1	38.0	22.4	23.5	13.8
FLOOR-4	85	40	18	2.5	25	MHS	OK-N-F	TWO	2	19.0	17.9	17.2	11.1
FLOOR-5	85	35	18	3.3	29	MHS	OK-N-F	TWO	2	19.0	15.7	17.2	9.7
FLOOR-6	85	30	18	4.5	34	SLN	OK-N-F	TWO	2	19.0	13.4	17.2	8.3
								CENTRAL	1	38.0	13.4	23.5	8.3
FLOOR-7	75	65	15	1.1	18	MHS	OK-N-F	CENTRAL	1	33.5	29.1	20.7	18.0
FLOOR-8	75	55	15	1.5	21	MHS	OK-N-F	CENTRAL	1	33.5	24.6	20.7	15.2
FLOOR-9	75	50	15	1.8	23	MHS	OK-N-F	CENTRAL	1	33.5	22.4	20.7	13.8
FLOOR-10	75	40	18	2.8	28	MHS	OK-N-F	CENTRAL	1	33.5	17.9	20.7	11.1
FLOOR-11	75	35	18	3.8	33	MHS	OK-N-F	TWO	2	16.8	15.7	15.2	9.7
FLOOR-12	75	30	18	5.2	39	SLN	OK-N-F	TWO	2	16.8	13.4	15.2	8.3
								CENTRAL	1	33.5	13.4	20.7	8.3
FLOOR-13	65	65	15	1.2	20	MHS	OK-N-F	CENTRAL	1	29.1	29.1	18.0	18.0
FLOOR-14	65	55	18	1.7	24	MHS	OK-N-F	CENTRAL	1	29.1	24.6	18.0	15.2
FLOOR-15	65	50	18	2.1	26	MHS	OK-N-F	CENTRAL	1	29.1	22.4	18.0	13.8
FLOOR-16	65	40	18	3.4	34	MHS	OK-N-F	CENTRAL	1	29.1	17.9	18.0	11.1
FLOOR-17	65	35	18	4.3	38	MHS	OK-N-F	CENTRAL	1	29.1	15.7	18.0	9.7
FLOOR-18	65	30	18	5.9	44	SLN	OK-N-F	TWO	2	14.5	13.4	13.1	8.3
								CENTRAL	1	29.1	13.4	18.0	8.3
FLOOR-19	55	55	18	2.0	28	MHS	OK-N-F	CENTRAL	1	24.6	24.6	15.2	15.2
FLOOR-20	55	50	18	2.5	31	MHS	OK-N-F	CENTRAL	1	24.6	22.4	15.2	13.8
FLOOR-21	55	40	18	3.9	39	MHS	OK-N-F	CENTRAL	1	24.6	17.9	15.2	11.1
FLOOR-22	55	35	18	5.1	45	SLN	OK-N-F	CENTRAL	1	24.6	15.7	15.2	9.7
FLOOR-24	45	40	18	4.8	48	SLN	OK-N-F	CENTRAL	1	20.1	17.9	12.4	11.1

Part-E: Evaluation Preferences Supplied by the Designer

Flexibility of rentable areas	100
Window line for rentable areas	100
Suitability for lateral structural system	50
High rentability at ground level	0
Travel distance from core to window-line	100
Clarity of circulation of rental areas	50
Daylight and view for core areas	0
Service connections at roof	0
Service connections at ground	0
General Energy Efficiency	100

Part-F: Floor Evaluation Result

(Not Sorted by Rank)

Floor Plan	#Floors	Core-Type	Eval-Value	Rank
FLOOR-1	16	CENTRAL	2050.0	2
FLOOR-2	19	CENTRAL	1950.0	3
FLOOR-3	20	CENTRAL	1950.0	3
FLOOR-4	25	TWO	1650.0	5
FLOOR-5	29	TWO	1650.0	5
FLOOR-6	34	CENTRAL	1850.0	4
FLOOR-6	34	TWO	1650.0	5
FLOOR-7	18	CENTRAL	2050.0	2
FLOOR-8	21	CENTRAL	2050.0	2
FLOOR-9	23	CENTRAL	2050.0	2
FLOOR-10	28	CENTRAL	1950.0	3
FLOOR-11	33	TWO	1650.0	5
FLOOR-12	39	CENTRAL	1850.0	4
FLOOR-12	39	TWO	1650.0	5
FLOOR-13	20	CENTRAL	2150.0	1
FLOOR-14	24	CENTRAL	2050.0	2
FLOOR-15	26	CENTRAL	2050.0	2
FLOOR-16	34	CENTRAL	1950.0	3
FLOOR-17	38	CENTRAL	1950.0	3
FLOOR-18	44	CENTRAL	1850.0	4
FLOOR-18	44	TWO	1650.0	5
FLOOR-19	28	CENTRAL	2150.0	1
FLOOR-20	31	CENTRAL	2050.0	2
FLOOR-21	39	CENTRAL	2050.0	2
FLOOR-22	45	CENTRAL	1950.0	3
FLOOR-24	48	CENTRAL	2050.0	2

SUMMARY OF STRUCURAL LAYOUT ALTERNATIVES

(Note: Not all alternatives described have a corresponding graphical representation. Only a representative few are presentaed)

Alternative Structural Schemes for Layout#22

Structural Scheme Designation: Fr-Shearwl-Scheme-10

```
Structure Type: Frame ShearWall Scheme
Major Stability Element: Frame-Shearwall
Column LayoutType: As-Per-Shearwalls
  Clear Span between core and window line:
      Front & Back edges to Core: 9.7m
  Clear Span between core and window line:
      Left & Right edges to Core: 15.2m
  Note: 2 span(s) of Girders at Left and Right Ends each of 7.6m
 1.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):5.8m
 2.Column Spacing:
    Perimeter (Front & Back):6.9m
    Perimeter (Sides):7.0m
 3.Column Spacing:
    Perimeter (Front & Back):4.6m
    Perimeter (Sides):4.4m
  Number of Shear Walls in Core:
    YY direction (North-South): 3
    XX direction (East-West):   2
        (Note: XX walls are the flanges of YY walls)
```

Structural Scheme Designation: Steel-Framed-Tube-Scheme-8

```
Structure Type: Framed Tube Scheme
Major Stability Element: Frame-Shearwall
Column LayoutType: As-Per-Shearwalls
  Clear Span between core and window line:
      Front & Back edges to Core: 9.7m
  Clear Span between core and window line:
      Left & Right edges to Core: 15.2m
  Note: 2 span(s) of Girders at Left and Right Ends each of 7.6m
 1.Column Spacing:
    Perimeter (Front & Back):3.1m
    Perimeter (Sides):2.9m
 2.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):5.8m
 3.Column Spacing:
    Perimeter (Front & Back):7.9m
    Perimeter (Sides):8.7m
 4.Column Spacing:
    Perimeter (Front & Back):4.6m
    Perimeter (Sides):4.4m
  Number of Shear Walls in Core:
    YY direction (North-South): 3
    XX direction (East-West):   2
        (Note: XX walls are the flanges of YY walls)
```

Structural Scheme Designation: Braced-Perimeter-5

```
Structure Type: Braced Frame Scheme
Major Stability Element: Braced-Frame-Perimeter
Column LayoutType: Perimeter-Based
  Clear Span between core and window line:
      Front & Back edges to Core: 9.7m
  Clear Span between core and window line:
      Left & Right edges to Core: 15.2m
  Note: 2 span(s) of Girders at Left and Right Ends each of 7.6m
 1.Column Spacing:
    Perimeter (Front & Back):9.2m
    Perimeter (Sides):8.7m
 2.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):5.8m
  Structural Grid inside the Core area: 6.1m x 5.2m
```

Structural Scheme Designation: Braced-Internal-6

```
Structure Type: Braced Frame Scheme
Major Stability Element: Braced-Frame-Internal
Column LayoutType: Grid-Aligned-To-Core
  Clear Span between core and window line:
      Front & Back edges to Core: 9.7m
  Clear Span between core and window line:
      Left & Right edges to Core: 15.2m
  Note: 2 span(s) of Girders at Left and Right Ends each of 7.6m
 1.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):5.8m
  Structural Grid inside the Core area: 6.1m x 5.2m
```

Approximate Column Sizes for Layout#22

Structural Scheme: Fr-Shearwl-Scheme-10

Column Layout Type: As-Per-Shearwalls

Shearwall Details:

```
Number of Shearwalls (with flanges) in Core: 3
Concrete strength: 55MPa
Spacing of Shearwalls:12.3m
1. Thickness of wall at base: 600mm
    Wall Shape: C
2. Thickness of wall at base: 600mm
    Wall Shape: I
3. Thickness of wall at base: 600mm
    Wall Shape: C-INVERTED
Shearwall concrete volume for this alternative:  4533. cu.m.
  All Walls may taper to 250mm at top;
  May be terminated in some cases;
  & also in cases where elevator banks drop off.
```

Details of Columns:

Alternative with Perimeter column spacings XX: 4.6m YY: 4.4m
CONCRETE Strength: 55MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	300	300	300
40-36	300	300	300
35-31	300	300	300
30-26	350	350	350
25-21	500	500	500
20-16	600	600	600
15-11	650	650	650
10- 6	700	700	700
5- 1	750	750	750

Column and Shearwall concrete volume for this alternative:6763. Cu.m.

Alternative with Perimeter column spacings XX: 6.9m YY: 7.0m
CONCRETE Strength: 55Mpa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	300	300	300
40-36	300	300	300
35-31	350	350	350
30-26	500	500	500
25-21	650	650	650
20-16	700	700	700
15-11	750	750	750
10- 6	800	800	800
5- 1	850	850	850

Column and Shearwall concrete volume for this alternative:6906. Cu.m.

Alternative with Perimeter column spacings XX: 6.1m YY: 5.8m
CONCRETE Strength: 55MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	300	300	300
40-36	300	300	300
35-31	350	350	350
30-26	500	500	500
25-21	600	600	600
20-16	700	700	700
15-11	750	750	750
10- 6	750	750	750
5- 1	800	800	800

Column and Shearwall concrete volume for this alternative:6940. Cu.m.

Structural Scheme: Steel-Framed-Tube-Scheme-8

Column Layout Type: As-Per-Shearwalls

Shearwall Details:

```
Number of Shearwalls (with flanges) in Core: 3
Concrete strength: 55MPa
Spacing of Shearwalls:12.3m
1. Thickness of wall at base: 600mm
    Wall Shape: C
2. Thickness of wall at base: 600mm
    Wall Shape: I
3. Thickness of wall at base: 600mm
    Wall Shape: C-INVERTED
Shearwall concrete volume for this alternative:  4533. cu.m.
  All Walls may taper to 250mm at top;
  May be terminated in some cases;
  & also in cases where elevator banks drop off.
```

Details of Columns:

Alternative with Perimeter column spacings XX: 4.6m   YY: 4.4m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W200 200)	(W200 200)	(W200 200)
40-36	(W250 250)	(W250 250)	(W250 250)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W310 310)	(W310 310)	(W310 310)
25-21	(W310 310)	(W310 310)	(W310 310)
20-16	(W360 360)	(W360 360)	(W360 360)
15-11	(W360 360)	(W360 360)	(W360 360)
10- 6	(W360 360)	(W360 360)	(W360 360)
5- 1	(W360 360)	(W360 360)	(W360 360)

Column Steel weight for this alternative: 378000.0 KG

Alternative with Perimeter column spacings XX: 7.9m   YY: 8.7m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W200 200)	(W200 200)	(W200 200)
40-36	(W310 310)	(W310 310)	(W310 310)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W310 310)	(W310 310)	(W310 310)
25-21	(W360 360)	(W360 360)	(W360 360)
20-16	(W460 280)	(W460 280)	(W460 280)
15-11	(W460 280)	(W460 280)	(W460 280)
10- 6	(W610 325)	(W610 325)	(W610 325)
5- 1	(W460 280)	(W460 280)	(W460 280)

Column Steel weight for this alternative: 261000.0 KG

Alternative with Perimeter column spacings XX: 6.1m YY: 5.8m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W200 200)	(W200 200)	(W200 200)
40-36	(W310 310)	(W310 310)	(W310 310)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W310 310)	(W310 310)	(W310 310)
25-21	(W360 360)	(W360 360)	(W360 360)
20-16	(W360 360)	(W360 360)	(W360 360)
15-11	(W360 360)	(W360 360)	(W360 360)
10- 6	(W460 280)	(W460 280)	(W460 280)
5- 1	(W360 360)	(W360 360)	(W360 360)

Column Steel weight for this alternative: 315000.0 KG

Alternative with Perimeter column spacings XX: 3.1m YY: 2.9m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W200 200)	(W200 200)	(W200 200)
40-36	(W200 200)	(W200 200)	(W200 200)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W310 310)	(W310 310)	(W310 310)
25-21	(W310 310)	(W310 310)	(W310 310)
20-16	(W310 310)	(W310 310)	(W310 310)
15-11	(W310 310)	(W310 310)	(W310 310)
10- 6	(W360 360)	(W360 360)	(W360 360)
5- 1	(W310 310)	(W310 310)	(W310 310)

Column Steel weight for this alternative: 504000.0 KG

Table D2: Summary of gravity system alternatives for Layout#22.

Construction Material	Type of Gravity System	Floor Depth OR Beam/slab size mm x mm	Relative Prefereance	Column Layout Type
Concrete Floor Systems		Structural Scheme: Fr-Shearwl-Scheme-10		
	One-Way Beam-Slab	1250 deep 160 slab	3	As-per-Shearwalls
	Joist Slab	1100deep 400 joists	1	As-per-Shearwalls
	Waffle Slab	300 slab	2	As-per-Shearwalls
	Band-Beam Slab	1050 band 210 slab	4	As-per-Shearwalls
	Hollow-Core Slab	380 slab	5	As-per-Shearwalls
Composite Steel Deck Floor Systems with Steel Beams		Structural Scheme: Steel-Framed-Tube-Scheme-8		
	Tapered Beam	280x570	3	As-per-Shearwalls
	Truss Beam	680deep +70deck	1	As-per-Shearwalls
	Haunch Beam	280x460	2	As-per-Shearwalls
	Parallel Beam	230x460	4	As-per-Shearwalls
	Stub Girder	280x570	5	As-per-Shearwalls

Note: 1 is most preferred; Based on historical in-place cost, on a relative basis. Ranking not based on actual quantities.

Sample Images of Alternatives for Layout#22

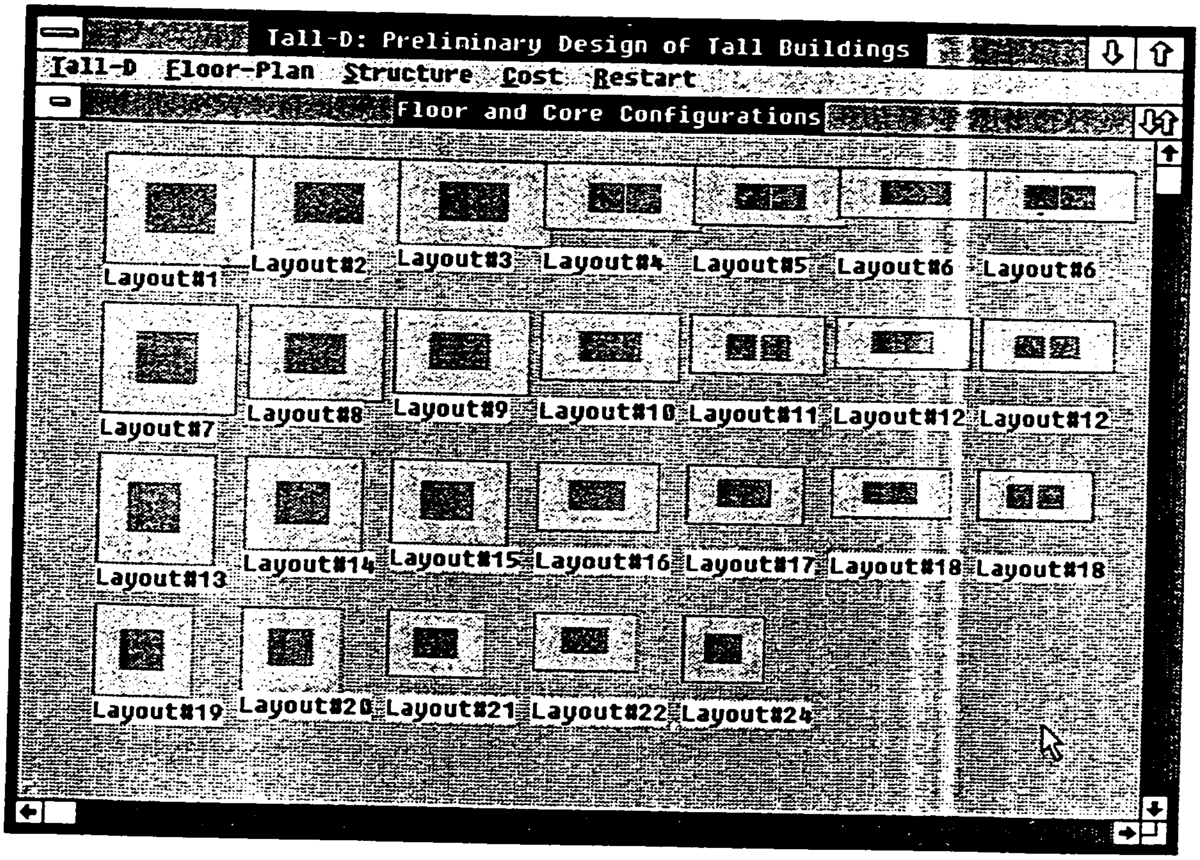


Fig. D2- 1 Floor layout alternatives retained at end of Phase-1 of design.

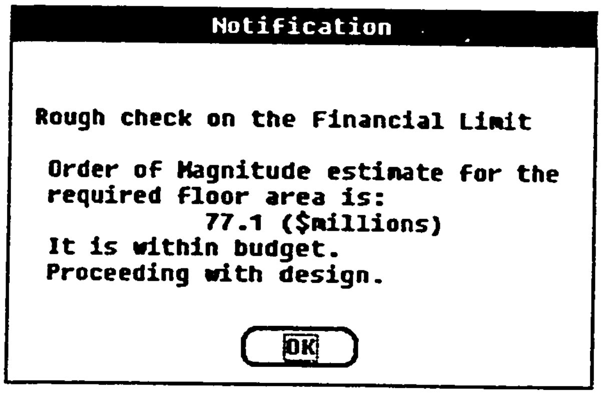


Fig. D2- 2 Initial check on cost

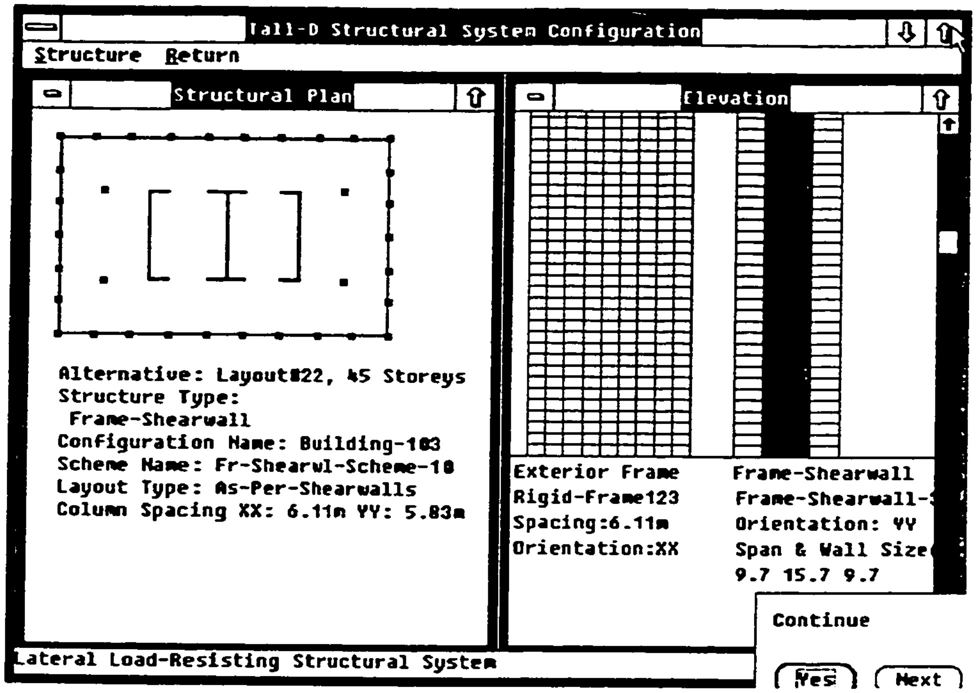


Fig. D2- 3Frame-shearwall strucural system alternative for Layout#22: Perimeter column spacing 6.11m and 5.83m

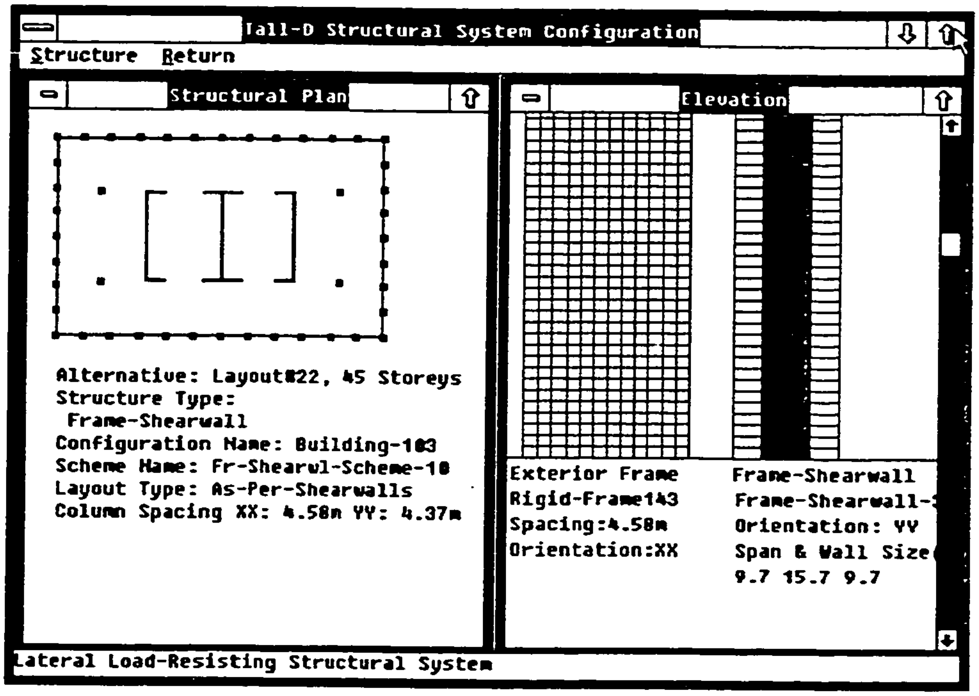


Fig. D2- 4 Frame-shearwall strucural system alternative for Layout#22: Perimeter column spacing 4.58m and 4.37m

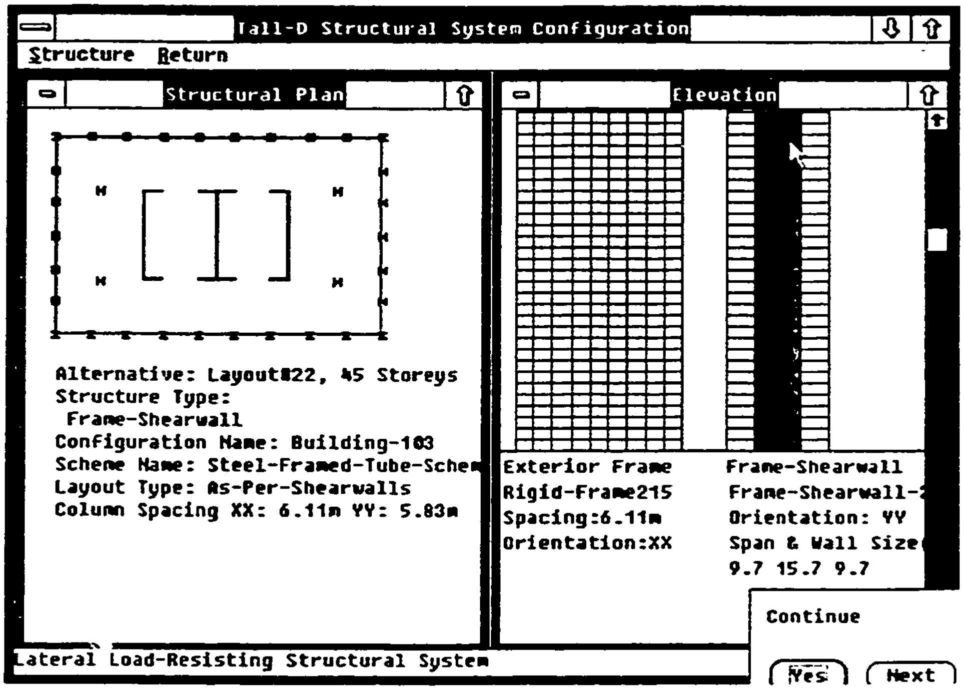


Fig. D2- 5 Steel Frame-shearwall strucural system alternative for Layout#22: Perimeter column spacing 6.11m and 5.83m

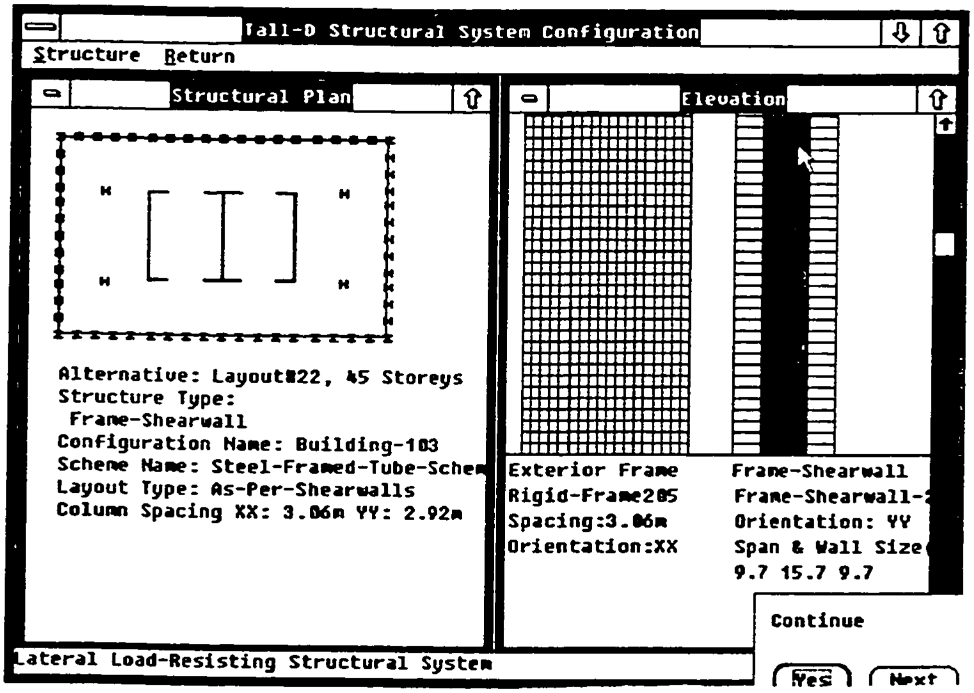


Fig. D2- 6 Steel Frame-shearwall strucural system alternative for Layout#22: Perimeter column spacing 3.06m and 2.92m

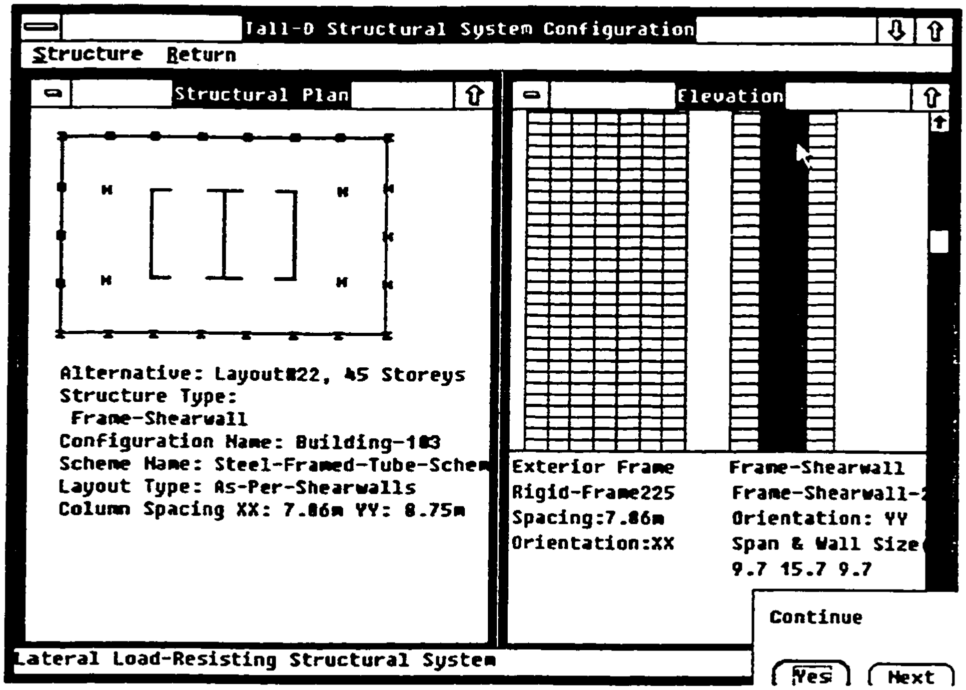


Fig. D2- 7 Steel Frame-shearwall strucural system alternative for Layout#22: Perimeter column spacing 7.86m and 8.75m

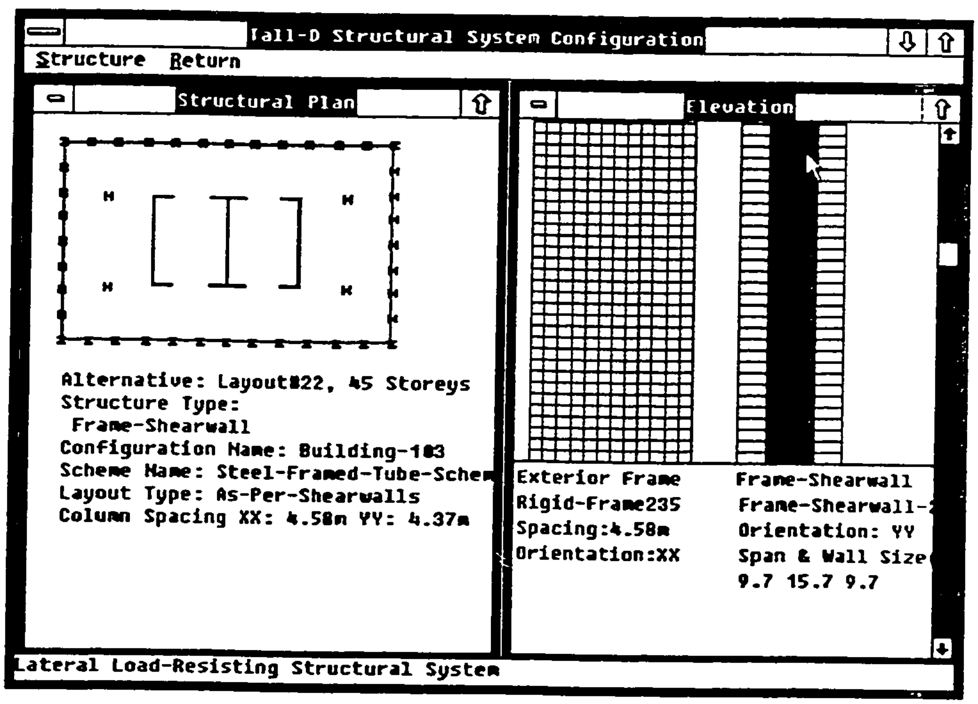


Fig. D2- 8 Steel Frame-shearwall strucural system alternative for Layout#22: Perimeter column spacing 4.58m and 4.37m

APPENDIX D-3: Design Case 3 - 1000 de La Gauchetière

TABLE OF CONTENTS

TABLE OF FIGURES

Overall Configuration

Part-A: General Design Requirements

Name of the Building	DE-LE-GAUCHETIERE
Length of Plot*	84 m
Width of Plot	105 m
Finance Limit	$150 million
Max. no. of Floors	60
Required Floor Area	100000 m^2

Part-B: Initial Floor Outlines Considered

Outline	Length	Width	Perm-FAR	Status
FLOOR-1	75	75	12	To-delete: F-A-R
FLOOR-2	75	65	15	OK-FAR
FLOOR-3	75	55	15	OK-FAR
FLOOR-4	75	50	15	OK-FAR
FLOOR-5	75	40	18	OK-FAR
FLOOR-6	75	35	18	OK-FAR
FLOOR-7	75	30	18	OK-FAR
FLOOR-8	65	65	15	OK-FAR
FLOOR-9	65	55	18	OK-FAR
FLOOR-10	65	50	18	OK-FAR
FLOOR-11	65	40	18	OK-FAR
FLOOR-12	65	35	18	OK-FAR
FLOOR-13	65	30	18	OK-FAR
FLOOR-14	55	55	18	OK-FAR
FLOOR-15	55	50	18	OK-FAR
FLOOR-16	55	40	18	OK-FAR
FLOOR-17	55	35	18	OK-FAR
FLOOR-18	55	30	18	OK-FAR
FLOOR-19	45	40	18	OK-FAR
FLOOR-20	45	35	18	OK-FAR
FLOOR-21	45	30	18	OK-FAR
FLOOR-22	40	40	18	OK-FAR
FLOOR-23	40	35	18	OK-FAR
FLOOR-24	40	30	18	OK-FAR

* The way Tall-D is implemented, length is the dimension parallel to the street with main entrance to the building, which is 84m here. This allows for input of rectangular plots of any proportion.

Part-C: Status at the end of Level-I

Outline	Height(m)	BSR	#-Fl	Mch-Flrs	Status
FLOOR-1	NIL	NIL	NIL	NIL	To-delete: F-A-R
FLOOR-2	100.0	1.5	25	1	OK-N-F
FLOOR-3	120.0	2.2	30	1	OK-N-F
FLOOR-4	132.0	2.6	33	2	OK-N-F
FLOOR-5	164.0	4.1	41	2	OK-N-F
FLOOR-6	188.0	5.4	47	2	OK-N-F
FLOOR-7	216.0	7.2	54	2	OK-N-F
FLOOR-8	116.0	1.8	29	1	OK-N-F
FLOOR-9	140.0	2.5	35	2	OK-N-F
FLOOR-10	152.0	3.0	38	2	OK-N-F
FLOOR-11	188.0	4.7	47	2	OK-N-F
FLOOR-12	216.0	6.2	54	2	OK-N-F
FLOOR-13	NIL	NIL	62	NIL	DEL-N-F
FLOOR-14	164.0	3.0	41	2	OK-N-F
FLOOR-15	180.0	3.6	45	2	OK-N-F
FLOOR-16	220.0	5.5	55	2	OK-N-F
FLOOR-17	NIL	NIL	64	NIL	DEL-N-F
FLOOR-18	NIL	NIL	74	NIL	DEL-N-F
FLOOR-19	NIL	NIL	68	NIL	DEL-N-F
FLOOR-20	NIL	NIL	78	NIL	DEL-N-F
FLOOR-21	NIL	NIL	90	NIL	DEL-N-F
FLOOR-22	NIL	NIL	77	NIL	DEL-N-F
FLOOR-23	NIL	NIL	87	NIL	DEL-N-F
FLOOR-24	NIL	NIL	101	NIL	DEL-N-F

Part-D: Core Configuration

Outline	FL-L	FL-W	F-A-R	B-S-R	#-Flrs	SLNDR	Status	CORE	#-CORES	CORE-L	CORE-W	CORE-X	CORE-Y
FLOOR-2	75	65	15	1.5	25	MHS	OK-N-F	CENTRAL	1	29.0	25.2	23.0	19.9
FLOOR-3	75	55	15	2.2	30	MHS	OK-N-F	CENTRAL	1	29.0	21.3	23.0	16.8
FLOOR-4	75	50	15	2.6	33	MHS	OK-N-F	CENTRAL	1	29.0	19.4	23.0	15.3
FLOOR-5	75	40	18	4.1	41	MHS	OK-N-F	CENTRAL	1	29.0	15.5	23.0	12.3
FLOOR-6	75	35	18	5.4	47	SLN	OK-N-F	TWO	2	14.5	13.6	18.2	10.7
								CENTRAL	1	29.0	13.6	23.0	0.7
FLOOR-7	75	30	18	7.2	54	VSLN	OK-N-F	ENDS	2	5.6	30.0	0.0	0.0
FLOOR-8	65	65	15	1.8	29	MHS	OK-N-F	CENTRAL	1	25.2	25.2	19.9	19.9
FLOOR-9	65	55	18	2.5	35	MHS	OK-N-F	CENTRAL	1	25.2	21.3	19.9	16.8
FLOOR-10	65	50	18	3.0	38	MHS	OK-N-F	CENTRAL	1	25.2	19.4	19.9	15.3
FLOOR-11	65	40	18	4.7	47	SLN	OK-N-F	CENTRAL	1	25.2	15.5	19.9	12.3
FLOOR-12	65	35	18	6.2	54	VSLN	OK-N-F	CENTRAL	1	25.2	13.6	19.9	10.7
FLOOR-14	55	55	18	3.0	41	MHS	OK-N-F	CENTRAL	1	21.3	21.3	16.8	16.8
FLOOR-15	55	50	18	3.6	45	MHS	OK-N-F	CENTRAL	1	21.3	19.4	16.8	15.3
FLOOR-16	55	40	18	5.5	55	SLN	OK-N-F	CENTRAL	1	21.3	15.5	16.8	12.3

Part-E: Evaluation Preferences Supplied by the Designer

Flexibility of rentable areas	100
Window line for rentable areas	100
Suitability for lateral structural system	50
High rentability at ground level	0
Travel distance from core to window-line	100
Clarity of circulation of rental areas	100
Daylight and view for core areas	0
Service connections at roof	0
Service connections at ground	100
General Energy Efficiency	100

Part-F: Floor Evaluation Result

(Not Sorted by Rank)

Floor Plan	#Floors	Core-Type	Eval-Value	Rank
FLOOR-2	25	CENTRAL	2650.0	2
FLOOR-3	30	CENTRAL	2650.0	2
FLOOR-4	33	CENTRAL	2650.0	2
FLOOR-5	41	CENTRAL	2550.0	3
FLOOR-6	47	CENTRAL	2450.0	4
FLOOR-6	47	TWO	2250.0	5
FLOOR-7	54	ENDS	1600.0	6
FLOOR-8	29	CENTRAL	2750.0	1
FLOOR-9	35	CENTRAL	2650.0	2
FLOOR-10	38	CENTRAL	2650.0	2
FLOOR-11	47	CENTRAL	2550.0	3
FLOOR-12	54	CENTRAL	2550.0	3
FLOOR-14	41	CENTRAL	2750.0	1
FLOOR-15	45	CENTRAL	2650.0	2
FLOOR-16	55	CENTRAL	2650.0	2

SUMMARY OF STRUCURAL LAYOUT ALTERNATIVES

(Note: Not all alternatives described have a corresponding graphical representation. Only a representative few are presentaed)

Alternative Structural Schemes for Layout#15

Structural Scheme Designation: Fr-Shearwl-Scheme-10

```
Structure Type: Frame ShearWall Scheme
Major Stability Element: Frame-Shearwall
Column LayoutType: As-Per-Shearwalls
  Clear Span between core and window line:
      Front & Back edges to Core: 15.3m
  Clear Span between core and window line:
      Left & Right edges to Core: 16.8m
 1.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):6.2m
 2.Column Spacing:
    Perimeter (Front & Back):6.9m
    Perimeter (Sides):7.1m
 3.Column Spacing:
    Perimeter (Front & Back):4.6m
    Perimeter (Sides):4.5m
  Number of Shear Walls in Core:
    YY direction (North-South): 3
    XX direction (East-West):   3
       (Note: XX walls are the flanges of YY walls)
```

Structural Scheme Designation: Steel-Framed-Tube-Scheme-8

```
Structure Type: Framed Tube Scheme
Major Stability Element: Frame-Shearwall
Column LayoutType: As-Per-Shearwalls
  Clear Span between core and window line:
      Front & Back edges to Core: 15.3m
  Clear Span between core and window line:
      Left & Right edges to Core: 16.8m
 1.Column Spacing:
    Perimeter (Front & Back):3.1m
    Perimeter (Sides):2.9m
 2.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):6.2m
 3.Column Spacing:
    Perimeter (Front & Back):7.9m
    Perimeter (Sides):8.3m
 4.Column Spacing:
    Perimeter (Front & Back):4.6m
    Perimeter (Sides):4.5m
  Number of Shear Walls in Core:
    YY direction (North-South): 3
    XX direction (East-West):   3
       (Note: XX walls are the flanges of YY walls)
```

Structural Scheme Designation: Braced-Perimeter-5

```
Structure Type: Braced Frame Scheme
Major Stability Element: Braced-Frame-Perimeter
Column LayoutType: Perimeter-Based
  Clear Span between core and window line:
      Front & Back edges to Core: 15.3m
  Clear Span between core and window line:
      Left & Right edges to Core: 16.8m
 1.Column Spacing:
    Perimeter (Front & Back):9.2m
    Perimeter (Sides):8.3m
 2.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):6.2m
  Structural Grid inside the Core area: 5.3m x 6.5m
```

Structural Scheme Designation: Braced-Internal-6

```
Structure Type: Braced Frame Scheme
Major Stability Element: Braced-Frame-Internal
Column LayoutType: Grid-Aligned-To-Core
  Clear Span between core and window line:
      Front & Back edges to Core: 15.3m
  Clear Span between core and window line:
      Left & Right edges to Core: 16.8m
 1.Column Spacing:
    Perimeter (Front & Back):6.1m
    Perimeter (Sides):6.2m
  Structural Grid inside the Core area: 5.3m x 6.5m
```

Approximate Column Sizes: Layout#15

(Note: Not all alternatives described have a corresponding graphical representation. Only a representative few are presentaed)

Structural Scheme: Fr-Shearwl-Scheme-10

Column Layout Type: As-Per-Shearwalls

Shearwall Details:

Number of Shearwalls (with flanges) in Core: 3
Concrete strength: 55MPa
Spacing of Shearwalls:10.7m
1. Thickness of wall at base: 675mm
   Wall Shape: C
2. Thickness of wall at base: 600mm
   Wall Shape: I
3. Thickness of wall at base: 675mm
   Wall Shape: C-INVERTED

Shearwall concrete volume for this alternative: 5556. cu.m.
All Walls may taper to 250mm at top;
May be terminated in some cases;
& also in cases where elevator banks drop off.

Details of Columns:

Alternative with Perimeter column spacings XX: 4.6m YY: 4.5m
CONCRETE Strength: 55MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	300	300	300
40-36	300	300	300
35-31	350	350	350
30-26	500	500	500
25-21	650	650	650
20-16	700	700	700
15-11	750	750	750
10- 6	800	800	800
5- 1	850	850	850

Column and Shearwall concrete volume for this alternative: 9014 cu.m.

Alternative with Perimeter column spacings XX: 6.9m YY: 7.1m
CONCRETE Strength: 55MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	300	300	300
40-36	350	350	350
35-31	500	500	500
30-26	700	700	700
25-21	750	750	750
20-16	800	800	800
15-11	850	850	850
10- 6	900	900	900
5- 1	900	900	900

Column and Shearwall concrete volume for this alternative:9064cu.m.

Alternative with Perimeter column spacings XX: 6.1m YY: 6.2m
CONCRETE Strength: 55MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	300	300	300
40-36	350	350	350
35-31	500	500	500
30-26	650	650	650
25-21	700	700	700
20-16	750	750	750
15-11	800	800	800
10- 6	900	900	900
5- 1	900	900	900

Column and Shearwall concrete volume for this alternative:9082cu.m.

Structural Scheme: Steel-Framed-Tube-Scheme-8

Column Layout Type: As-Per-Shearwalls

Shearwall Details:

```
Number of Shearwalls (with flanges) in Core: 3
Concrete strength: 55Mpa
Spacing of Shearwalls:10.7m
1. Thickness of wall at base: 675mm
    Wall Shape: C
2. Thickness of wall at base: 600mm
    Wall Shape: I
3. Thickness of wall at base: 675mm
    Wall Shape: C-INVERTED
Shearwall concrete volume for this alternative:  5556. cu.m.
  All Walls may taper to 250mm at top;
  May be terminated in some cases;
  & also in cases where elevator banks drop off.
```

Details of Columns:

Alternative with Perimeter column spacings XX: 4.6m   YY: 4.5m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W200 200)	(W200 200)	(W200 200)
40-36	(W310 310)	(W310 310)	(W310 310)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W310 310)	(W310 310)	(W310 310)
25-21	(W360 360)	(W360 360)	(W360 360)
20-16	(W360 360)	(W360 360)	(W360 360)
15-11	(W460 280)	(W460 280)	(W460 280)
10- 6	(W460 280)	(W460 280)	(W460 280)
5- 1	(W460 280)	(W460 280)	(W460 280)

Column Steel weight for this alternative: 432000.0 KG

Alternative with Perimeter column spacings XX: 7.9m   YY: 8.3m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W250 250)	(W250 250)	(W250 250)
40-36	(W310 310)	(W310 310)	(W310 310)
35-31	(W360 360)	(W360 360)	(W360 360)
30-26	(W360 360)	(W360 360)	(W360 360)
25-21	(W460 280)	(W460 280)	(W460 280)
20-16	(W610 325)	(W610 325)	(W610 325)
15-11	(W610 325)	(W610 325)	(W610 325)
10- 6	(W610 325)	(W610 325)	(W610 325)
5- 1	(W610 325)	(W610 325)	(W610 325)

Column Steel weight for this alternative: 297000.0 KG

Alternative with Perimeter column spacings XX: 6.1m YY: 6.2m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W250 250)	(W250 250)	(W250 250)
40-36	(W310 310)	(W310 310)	(W310 310)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W360 360)	(W360 360)	(W360 360)
25-21	(W360 360)	(W360 360)	(W360 360)
20-16	(W460 280)	(W460 280)	(W460 280)
15-11	(W610 325)	(W610 325)	(W610 325)
10- 6	(W610 325)	(W610 325)	(W610 325)
5- 1	(W460 280)	(W460 280)	(W460 280)

Column Steel weight for this alternative: 351000.0 KG

Alternative with Perimeter column spacings XX: 3.1m YY: 2.9m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W200 200)	(W200 200)	(W200 200)
40-36	(W250 250)	(W250 250)	(W250 250)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W310 310)	(W310 310)	(W310 310)
25-21	(W310 310)	(W310 310)	(W310 310)
20-16	(W360 360)	(W360 360)	(W360 360)
15-11	(W360 360)	(W360 360)	(W360 360)
10- 6	(W360 360)	(W360 360)	(W360 360)
5- 1	(W360 360)	(W360 360)	(W360 360)

Column Steel weight for this alternative: 594000.0 KG

Structural Scheme: Braced-Internal-6

Column Layout Type: Grid-Aligned-To-Core

Details of Columns:

Alternative with Perimeter column spacings XX: 6.1m YY: 6.2m
STEEL Strength: 220MPa

Storeys in Group	PerimeterXX (mm)	PerimeterYY (mm)	Corner (mm)
45-41	(W250 250)	(W250 250)	(W250 250)
40-36	(W310 310)	(W310 310)	(W310 310)
35-31	(W310 310)	(W310 310)	(W310 310)
30-26	(W360 360)	(W360 360)	(W360 360)
25-21	(W360 360)	(W360 360)	(W360 360)
20-16	(W460 280)	(W460 280)	(W460 280)
15-11	(W610 325)	(W610 325)	(W610 325)
10- 6	(W610 325)	(W610 325)	(W610 325)
5- 1	(W460 280)	(W460 280)	(W460 280)

Column Steel weight for the above alternative: 351000.0 KG

Table D3 Summary of gravity system alternatives for Layout#15.

Construction Material	Type of Gravity System	Floor Depth OR Beam/slab size mm x mm	Relative Preference	Column Layout Type
Concrete Floor Systems		**Structural Scheme: Fr-Shearwl-Scheme-10**		
	One-Way Beam-Slab	1400 deep 160 slab	3	As-per-Shearwalls
	Joist Slab	1250deep 410 joists	1	As-per-Shearwalls
	Waffle Slab	300 slab	2	As-per-Shearwalls
	Band-Beam Slab	1150 band 220 slab	4	As-per-Shearwalls
	Hollow-Core Slab	420 slab	5	As-per-Shearwalls
Composite Steel Deck Floor Systems with Steel Beams		**Structural Scheme: Steel-Framed-Tube-Scheme-8**		
	Tapered Beam	310x630	3	As-per-Shearwalls
	Truss Beam	750deep +70deck	1	As-per-Shearwalls
	Haunch Beam	310x500	2	As-per-Shearwalls
	Parallel Beam	250x500	4	As-per-Shearwalls
	Stub Girder	310x630	5	As-per-Shearwalls

Note: 1 is most preferred; Based on historical in-place cost, on a relative basis. Ranking not based on actual quantities.

Sample Images of Alternative Building Configurations

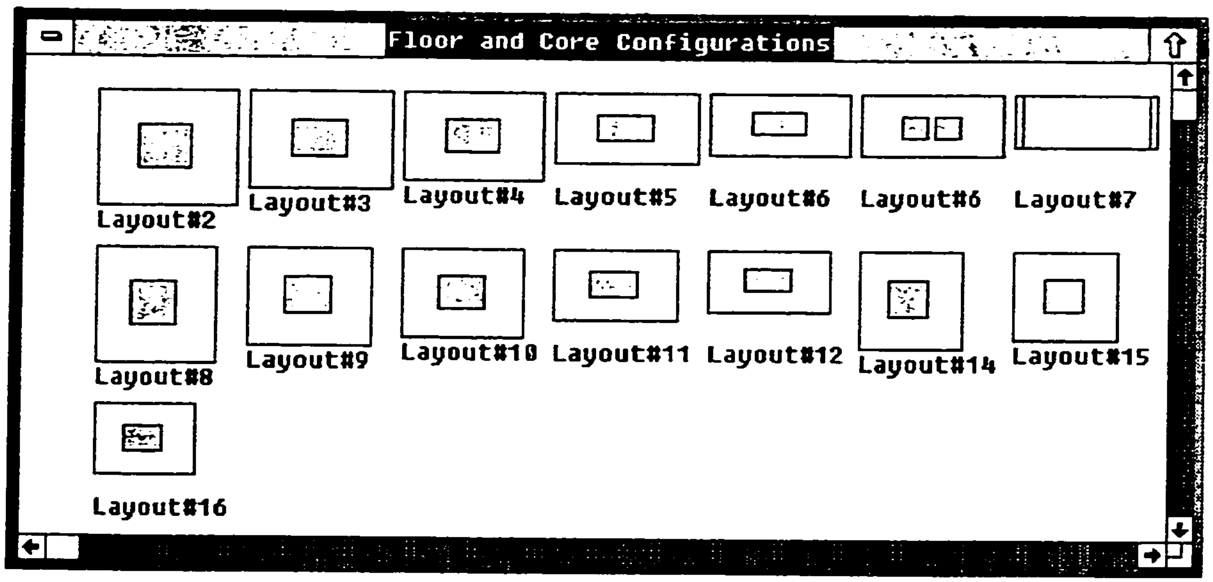


Fig. D3- 1Floor layout alternatives retained at end of Phase-1 of design.

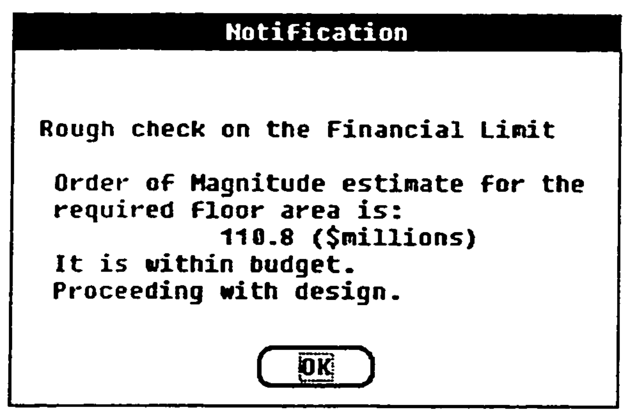


Fig. D3- 2 Initial check on cost

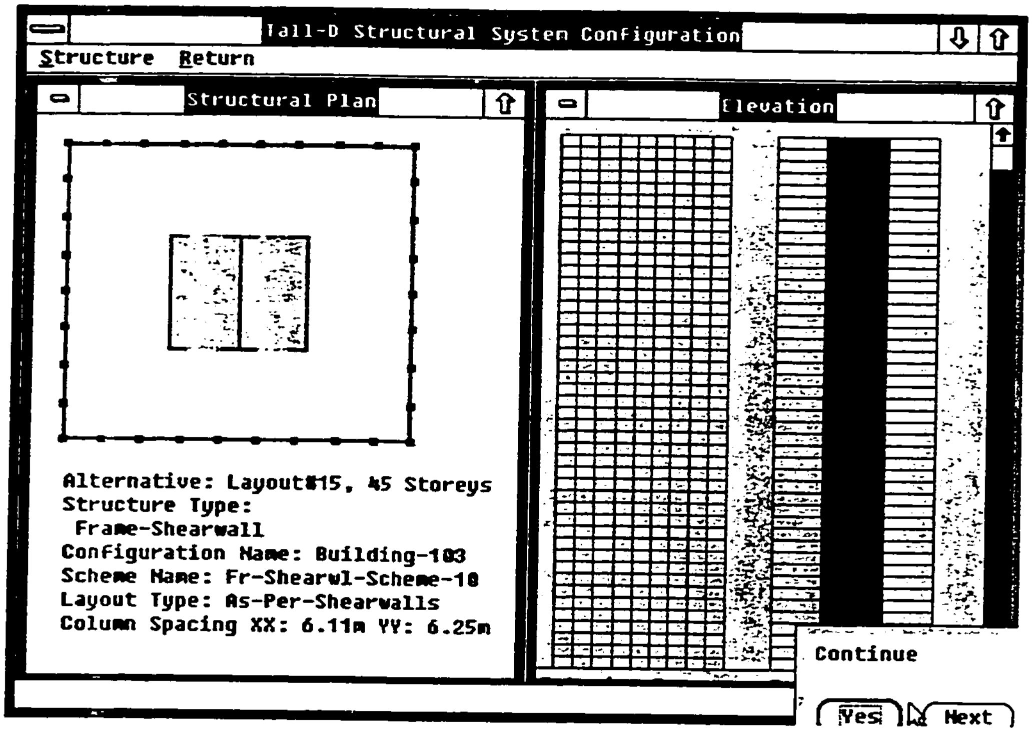


Fig. D3- 3 Frame-shearwall strucural system alternative for Layout#15: Perimeter column spacing 6.11m and 6.25m

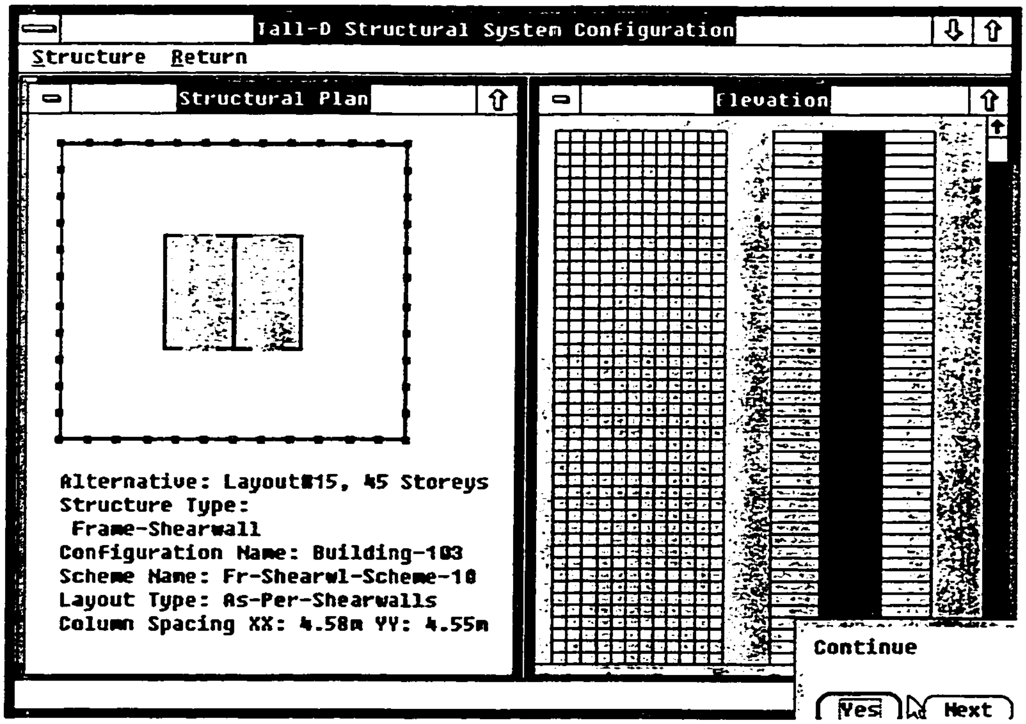


Fig. D3- 4 Frame-shearwall strucural system alternative for Layout#15: Perimeter column spacing 4.58m and 4.55m

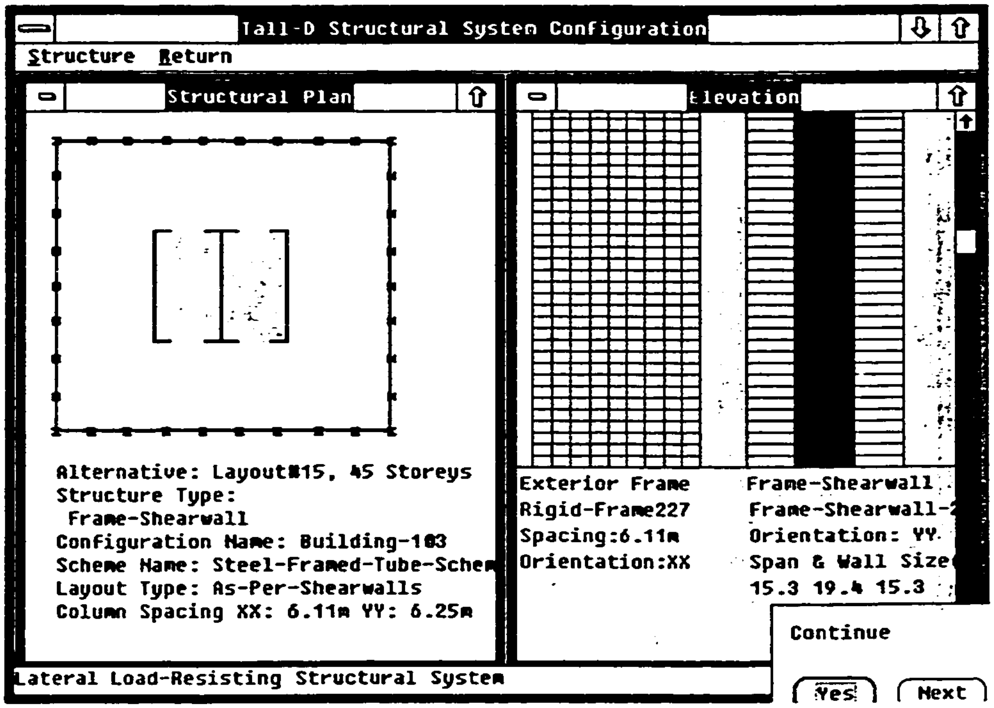


Fig. D3- 5 Steel framed-tube strucural system alternative for Layout#15: Perimeter column spacing 6.11m and 6.25m.

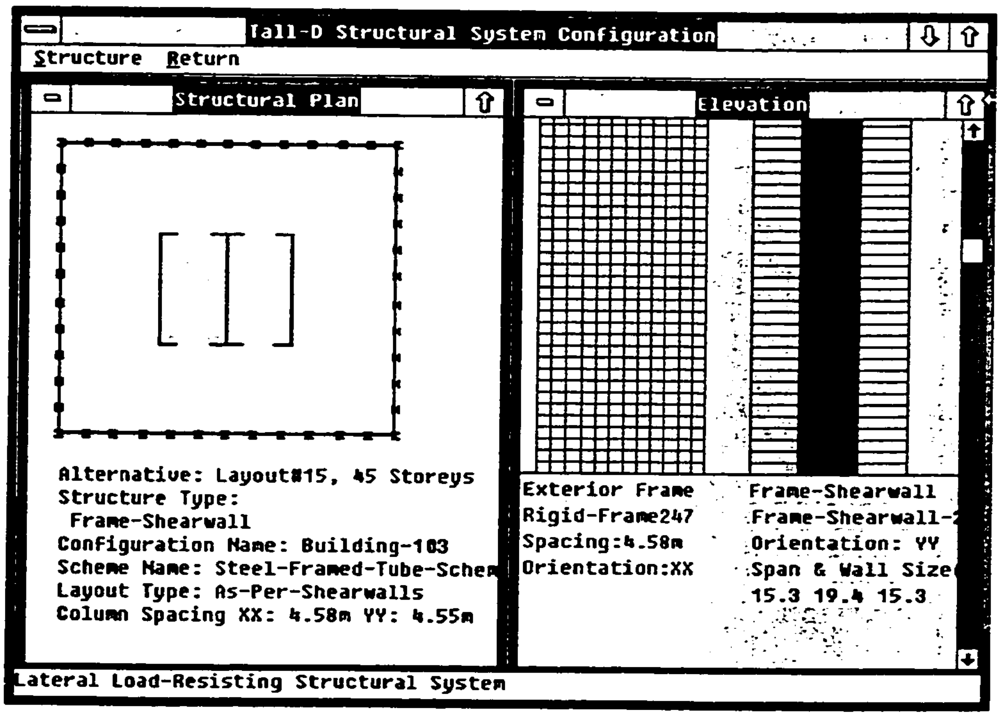


Fig. D3- 6 Steel framed-tube strucural system alternative for Layout#15: Perimeter column spacing 4.58m and 4.55m.

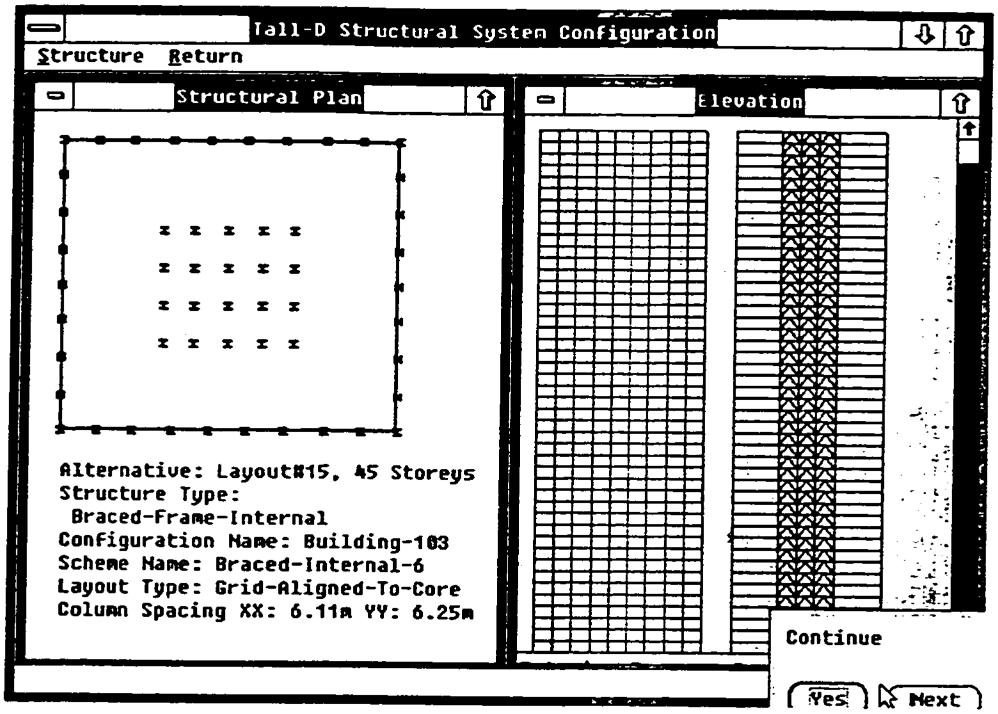


Fig. D3- 7 Steel braced-frame (internal) strucural system alternative for Layout#15: Perimeter column spacing 6.11m and 6.25m

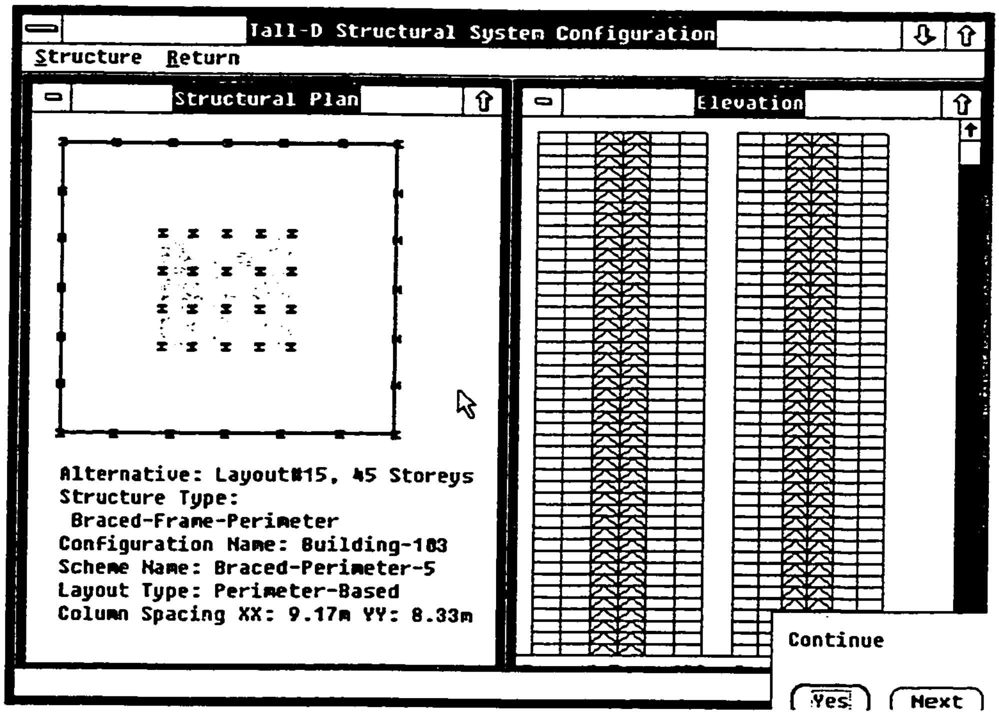


Fig. D3- 8 Steel braced-frame (on perimeter) strucural system alternative for Layout#15: Perimeter column spacing 9.17m and 8.33m